ABSTRACT

RAMACHANDRAN, VINAY. Bryce: An Extreme Programming (XP) Process Analysis and Project Management Tool (Under the direction of Dr. Laurie Williams.)

Increasingly, products in the software industry need to be delivered quickly to beat the competition. These products often have a dynamic and vague set of requirements. As a result, there has recently been a rapidly-rising interest in new approaches to software development. These approaches are lightweight and adaptive and have been called *agile* process methodologies. Extreme Programming (XP) is one such agile software process methodology.

XP is gaining popularity at a rapid rate, despite the lack of comprehensive, quantifiable studies demonstrating its effectiveness. In today's fast-paced business environments, project managers need to obtain information, track, and manage projects as quickly as possible. Though project management tools are commercially available, none are specifically designed for XP and cannot capture the information regarding the elements specific to XP e.g. user story information, refactoring information, etc. Moreover, existing tools tend to be labor-intensive to learn and use. Additionally, these existing tools are mainly geared toward traditional software processes.

This research involves the development and evaluation of a tool called *Bryce*. Bryce is a project management and process analysis tool. Bryce facilitates project management and communication within a team by means of data entry, providing elaborate reporting on project status and progress. This

data recorded in the tool can also be analyzed to provide invaluable quantitative information in assessing the XP methodology's effectiveness.

Bryce, a web-based tool, was developed using Java Server Pages (JSP). The tool was evaluated through its use in two graduate-level courses, CSC517 [Object-Oriented Languages and Systems, 49 students] and ECE 521/ECE 463 [Computer Design & Technology / Advanced Microprocessor Systems Design, 49 students], and one undergraduate course CSC326 [Software Engineering, 140 students] at North Carolina State University in the Fall 2002 semester. Furthermore, it has also been installed for managing XP projects at two industrial locations: International Business Machines (IBM) and Genesys Telecommunications Laboratories. Through this research, we observed the following:

1. Bryce appears to be an effective tool for managing XP projects.
2. Bryce may aid in research on XP by obtaining quantitative information regarding XP projects.

BRYCE: AN EXTREME PROGRAMMING (XP) PROCESS ANALYSIS AND PROJECT MANAGEMENT TOOL

by
VINAY RAMACHANDRAN

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

COMPUTER SCIENCE

Raleigh

2002

APPROVED BY:

Dr. Annie I. Antón Advisory Committee Member	Dr. Mladen A. Vouk Advisory Committee Member

Dr. Laurie Williams
Chair of Advisory Committee

Biography

Vinay Ramachandran received a Bachelor of Engineering degree in Computer Science in 1998 from B.M.S College of Engineering, Bangalore University (Bangalore, India). He worked in Wipro Technologies – Cisco Development Facility (Bangalore, India) for two years. He joined the M.S program at North Carolina State University in the year 2001. He has been a research assistant for Dr. Laurie Williams since May 2001.

Acknowledgements

I would first like to thank my advisor, Dr. Laurie Williams, who gave me an opportunity to do research in my field of interest, Extreme Programming. She has helped me in every aspect of this research - starting off from suggesting my thesis topic through the completion of my thesis. She provided me with all the required resources for this research and opportunities to attend XP Universe conferences in 2001 and 2002. Attending these conferences greatly helped me in my research. I could have not done my thesis without her keen insight, guidance and support. Finally, she got IBM and Genesys to use the tool, which was very important for the tool evaluation.

I would like to extend my appreciation to my committee Dr. Mladen Vouk and Dr. Annie Antón for their valuable inputs. I would like to thank the Center for Advance Computing and Communications (CACC) for funding this research. I would also like to thank Dr. Edward F Gehringer (North Carolina State University), Lucas Layman (North Carolina State University), Bill Krebs (IBM), and Patricia Gage (Genesys) who helped me with the evaluation of the tool.

Last but not the least, I would like to thank my parents and sister who have been my source of motivation and emotional strength. My M.S degree would not have been possible without their support.

Table of Contents

LIST OF TABLES VII

LIST OF FIGURES VIII

1 INTRODUCTION 1

1.1 Traditional Methodologies 1
1.2 Agile Software Methodologies 3
1.3 Extreme Programming 4
1.4 Hypotheses 9
1.5 Summary of Remaining Chapters 9

2 PROJECT MANAGEMENT 11

2.1 Introduction to Project Management 11
2.2 XP Project Management 12
2.3 Tools for XP project management 13
2.4 Collaboration and Discussion Tools 15
2.4.1 Conferencing 16
2.4.2 Whiteboarding 17
2.4.3 Wiki 18
2.5 Generic Project Management / Workflow Management Tools 20
2.5.1 MILOS ASE 21

3 INTRODUCTION TO BRYCE 24

3.1 Bryce Overview 24
3.2 List of features in Bryce 25
3.3 Bryce Documentation 26
3.4 Platform for Bryce Development and Access 27
3.5 Importance of Research 28
3.5.1 Project Management Capability 29
3.5.2 Research on XP 36
3.6 Benefits of using Bryce 41
3.7 Portability 42
3.8 Security 43

4 BRYCE FEATURES 46

4.1 Bryce Features Overview 46
4.2 New User Registration 47
4.3 Using Bryce 48
4.4 Description of Bryce menu items 48
4.4.1 My Stuff 49
4.4.2 Feature Management 50
4.4.3 Project Information 53
4.4.4 Administration 53

4.4.5 Manager Menu 54
4.4.6 Other Menu Items 55

5 PRELIMINARY BRYCE EVALUATION 56

5.1 METHOD FOR BRYCE EVALUATION 56
5.2 DETAILS ABOUT THE SURVEY 57
5.3 USER FEEDBACK 57
5.4 BRYCE FOR XP RESEARCH 59

6 CONCLUSION 61

6.1 FUTURE RESEARCH 62
6.1.1 Further tool Development 62

7 LIST OF REFERENCES 64

APPENDIX 1: INSTALLATION MANUAL 69

Requirements 69
Extracting the compressed tool archive 69
Installation and Configuration of Microsoft SQL Server 2000 70
Setting up of JDBC driver 71
Installation and Configuration of Resin 71
Installation and Configuration of IIS 72
Installation on Apache and Tomcat 73
Trouble Shooting 74

APPENDIX 2: USER MANUAL 75

Overview 75
New User Registration 76
Description of the Menu items 76
How do I use this Tool to track my Project? 82

APPENDIX 3: TUTORIAL 84

SETTING UP THE TOOL FOR YOUR PROJECT 84
7.1 INTRODUCTION 84
Getting Started 86
Requirements 86
Accessing the XP tool 87
New User Registration 87
Adding New Project (Manager Only) 90
Adding Release, Iteration (Manager Only) 92
Adding/Deleting New Team Member (Manager Only) 94
FEATURE MANAGEMENT 95
Adding a new story 95
Viewing and editing all stories 98
Adding a new Task 101

Viewing and editing all Tasks *103*
Adding a new Acceptance Test *104*
Viewing and editing all Acceptance Tests *105*
Recording time spent on a Task *106*
Editing the Recorded Task Time *110*
Changing Task status to completed *112*
Changing Story status to completed *114*
Making changes to a completed Story *118*
Keeping track of changes made to a story *122*
Viewing Project Report *124*
TOOL ADMINISTRATION 125
Add Team Member *125*
Delete User Registration *126*
Add/Change Manager *126*
Change Password *126*
APPENDIX 4: DATABASE DIAGRAM 127
APPENDIX 5: BRYCE SURVEY 128
PERSONAL INFORMATION 128
PROJECT MANAGEMENT 128
ABOUT STABILITY 130
ABOUT LEARNING, USAGE AND DOCUMENTATION 130
OVERVIEW 130

LIST OF TABLES

TABLE 2.4.1 COMPARISON OF WIKI AND BRYCE FEATURES.....20
TABLE 2.5.1 COMPARISON OF M-ASE AND BRYCE FEATURES.....23

LIST OF FIGURES

FIGURE 3.5.1. VIEWING ALL THE USER STORIES IN A PROJECT 30
FIGURE 3.5.2. VIEWING ALL THE USER TASKS IN A PROJECT 32
FIGURE 3.5.3. LISTING USER STORIES BELONGING TO THE CURRENT USER.....33
FIGURE 3.5.4. TRACKING THE REWORK DONE TO A USER STORY.....34
FIGURE 3.5.5. A SAMPLE PROJECT REPORT 35
FIGURE 3.5.6 THE COST OF CHANGE CURVE FOR TRADITIONAL PROCESSES (SOURCE [12]).....37
FIGURE 3.5.7 THE COST OF CHANGE CURVE FOR XP (SOURCE [12]).....38
FIGURE 3.5.8. REWORK INFORMATION OF A PROJECT 39
FIGURE 4.4.1. STORY TRAVERSING DIFFERENT STATES.....51

1 Introduction

This chapter provides an overview of the traditional methodologies for software development and the newer methodologies called agile methodologies. It discusses one of the most popular agile methodologies called Extreme Programming in more detail since it is served as the primary basis for this investigation. The hypothesis for this thesis is then presented. The summary of the remaining chapters is presented at the end of this chapter.

1.1 Traditional Methodologies

Developing reliable software on time and within budget is a difficult endeavor for many software organizations [1]. The waterfall model was introduced in a paper by W.W Royce [2] as a methodology for gaining control over software development and suggests a systematic, sequential approach to software development [3]. In the waterfall model, a project proceeds through the stages of requirements analysis, design, coding, testing and support with sign-off points at the end of each stage [3]. Some have found the application of the waterfall model problematic in more uncertain environments, such as Internet software engineering [4]. Uncertain environments with high requirements volatility necessitates greater interactivity between the customer and the development team [4].

To achieve the objective of greater interaction between the customer and the development team, several alternative software development models have

practices in which prototypes are developed and shown to customers early in the development process. The rapid-prototyping model [5] emphasizes the construction of an early prototype to help establish customer requirements [4]. Similarly, the spiral model [6] moves through a series of prototype builds to help the developers identify and reduce the risk associated with the project [4]. In both those models, however, the prototypes are not part of the design itself but merely representations that are thrown away after fulfilling their function as a communication vehicle. The bulk of the design work carried out thereafter is performed in a similar manner to the waterfall model [4]. In some projects, the time spent on a throwaway prototype typically does not result in code that can be reused in a full-scale system, and, hence, excessive effort spent on a throwaway prototype can be wasteful [7].

Another software development model, called evolutionary prototyping, involves the development of prototypes that are intended to evolve into the final version of the system [8]. The system partially developed as a prototype is demonstrated to the customer [8]. Continued development of the prototype is done based on customer feedback [8]. At some point the customer agrees that the prototype is “good enough” [8]. At this point, any remaining work on the system is completed and prototype is released as the final product [8]. The main drawback of this kind of prototyping is that it is not possible to know at the outset of the project how long it will take to create an acceptable product [8].

Alternatively, when the incremental software development model [3] is followed, a system is delivered to the customer in discrete chunks. This model is used as means of partitioning work so that some functionality can be delivered to the customer early [4].

In spite of these software development models, the success rate for software projects has been fairly poor [9]. A landmark 1994 white paper, "The CHAOS Report (1994)" [9] published by The Standish Group, a West Yarmouth, Massachusetts-based consultancy, reported that just 16 percent of software projects they studied succeeded. The rest either failed (31 percent) or were challenged (53 percent); a term encompassing cost and time overruns and missing features [9]. Another report by The Standish Group states that, “Underestimating project complexity and ignoring changing requirements are basic reasons why projects fail” [10].

1.2 Agile Software Methodologies

In an attempt to reduce the frequency of project failures, a new class of software development methodologies, called agile methodologies [11], have been introduced in the last five years. The most prominent difference between the traditional (waterfall, incremental) and agile methodologies is that the agile methodologies are less archival document-oriented [11]. Additionally, agile methods are designed to be adaptive rather than predictive [11]. Traditional methods promote planning out or predicting a large part of the software development activities in great detail for a long span of time, which works well

until the requirements change [11]. In an effort to respond to the changing requirements, agile methodologies use iterative development to incorporate or adapt to customer feedback throughout the development cycle, hence they are characterized as adaptive [11]. With agile planning, the long-term plans are very fluid, and the only stable plans are short-term plans that are made for a single iteration [11].

Some of the more popular agile methodologies are Extreme Programming (XP) [12], Dynamic Systems Development Method (DSDM) [13], Scrum [14], Crystal [15], Adaptive Software Development [16] and Feature Driven Development (FDD) [17]. Extreme Programming (XP) is considered the most popular of the agile methodologies [11, 18]. XP is also the most documented and specified of the agile methodologies, and its practices are appealing to many software developers. Chapter 3 discusses the development of a tool to support XP teams.

1.3 Extreme Programming

XP [12] is an agile methodology designed for small-to-medium sized teams developing software in the face of vague or rapidly changing requirements [12]. The roots of XP lie in the Smalltalk community and, in particular, in the close collaboration of Kent Beck and Ward Cunningham beginning in the late 1980's. Both of them refined these practices on numerous projects during the early 1990's, extending their ideas of a software development approach that was both adaptive and people-oriented [11]. The crucial step from informal practice

to a methodology occurred in 1997 when Kent Beck successfully used the full set of XP practices (which will be described later in this section) to implement a payroll project for Daimler Chrysler [11]. Since then, XP has been successfully used at many companies, such as Bayerische Landesbank, Credit Swiss Life, First Union National Bank, Ford Motor Company and UBS [19].

XP is based on Beck and Cunningham's observations of what made program development faster and what made it slower [20]. XP is an important new methodology because it is one of several new agile software methodologies created to produce high quality software [12] and reduce the cost of software [20]. XP is also professed to be a process that is simple and enjoyable [20].

Kent Beck [12] established the following set of four core values for XP:

- *Communication.* This XP value focuses on building a person-to-person, mutual understanding of the problem environment through minimal formal documentation and through maximum face-to-face interaction. XP's practices are designed to encourage interaction, developer-to-developer and developer-to-customer [12].
- *Simplicity.* This XP value challenges each team member to continuously ask, "What is the simplest thing that could possibly work?" [12]. The XP originators contend that it is better to do a simple thing today and pay a little more tomorrow for change than to do a more complicated thing today that may never be used [12].

- *Feedback.* XP teams obtain feedback by testing their software early and often [19]. They deliver the system to the customers as early as possible and implement changes and reprioritization as suggested by feedback from the customer.
- *Courage.* Developers often cite the pressure to ship a buggy product. It takes courage to resist this pressure [12]. Beck also states that XP teams must be courageous and willing to make changes late in the project or even discard the code and start all over again [12].

XP derives its practices from the above values. The 12 practices of XP are briefly discussed below [12]:

1. *The Planning Game*. The goal of the Planning Game is to quickly determine the scope of the next release by combining business priorities and technical estimates of product requirements.
2. *Small Releases*. A simple system is produced quickly, followed by new versions of increasing functionality in very short cycles of approximately three weeks.
3. *Metaphor*. A simple shared story of how the whole system works is used to guide all development.
4. *Simple design*. The system should be designed as simply as possible at any given moment. Extra complexity is removed as soon as it is discovered.

5. *Testing*. Programmers continually write automated unit tests, which must run flawlessly before development can continue. Customers also write acceptance tests, which are used to demonstrate that features have been implemented.
6. *Refactoring*. Programmers restructure the system without changing its behavior to remove duplication, improve communication, to simplify, and/or to add flexibility.
7. *Pair programming*. Pair programming is a style of programming in which two programmers work side by side at one computer, continually collaborating on the same design, algorithm, code or test [21, 22]. All production code in an XP project is written with two programmers at one machine.
8. *Collective Ownership*. Anyone can change any code anywhere in the system at anytime, regardless of who originally created the code.
9. *Continuous Integration*. The system is integrated and built several times each day, every time a task is completed.
10. *Sustainable Pace*. XP teams do not work an excessive number of hours per week for an extended period of time.
11. *On-site Customer*. An on-site customer should be available full time to answer questions for the development team.
12. *Coding standards*. This practice obligates the programmers to write all code in accordance to an established coding standard.

The XP practices are tightly coupled. XP originator Kent Beck asserts, “The practices support each other. The weakness of one is covered by the strengths of the others [12]”. Therefore, neglecting essential practices can have sub-optimal or negative consequences to team results. The practices that are actually utilized in team development can often be attributed to the perceptions of the value and difficulty of the practices by the developers on the team. Through two surveys answered by 27 developers, the utilization of and the sentiments towards the XP practices were assessed [23]. In general, all the practices were popular and well accepted with the exception of the metaphor practice [23]. Eighty six percent of the survey respondents indicated that their projects did not have metaphors. Among the most popular practices were planning games, pair programming and small releases [23]. Simple design was another highly popular practice of XP [23]. Seventy eight percent of respondents believed that simple design is the key to getting the job done quickly. Getting the job done quickly is not the only priority; it must also be done right [23]. However 33% of the respondents believed that simple design is not necessarily the right answer [23]. Perhaps these survey participants felt that complex design was inevitable under some circumstances [23]. Finally, another observation was that 92% of survey respondents believed that story cards were effective in capturing customer requirements. Participants were asked to rate whether XP helped them meet their objectives; they rated each of these aspects from –5 (Much worse) to 5

(Fully achieved). The participants felt that all of the following aspects were aided by XP:

- Deliver software in time (4)
- Develop software with a high quality, i.e. with less bugs (4)
- Let developers have fun with their work (3)
- Allow changes without incurring big costs (3)

Finally, almost all of them expressed that they would advocate using XP in the future [23].

1.4 Hypotheses

This research involved the development of Bryce, a web-based project management and process analysis tool designed for XP teams. Through this research, we examine the following two hypotheses in relation to this tool:

1. A lightweight project management tool will be beneficial for managing XP projects.
2. A lightweight measurement and analysis system will aid in research by obtaining quantitative information regarding XP projects.

1.5 Summary of Remaining Chapters

The remaining chapters of this thesis discuss XP project management and how Bryce can aid in the management of XP projects. Chapter 2 contains an overview of project management focusing on tools currently used for XP project management. Chapter 3 contains an explanation of the research motivation and

development of the Bryce tool. Chapter 4 contains an explanation of the features in the tool. Chapter 5 contains an explanation of how the tool was evaluated. Chapter 6 summarizes the conclusions and contributions of the thesis. It also suggests the areas for further development of the tool and research areas in which the tool could be used.

Several archival resources appear in the appendices. The Bryce installation manual, containing instructions for installing and setting up the tool, appears in Appendix 1. The Bryce user manual. containing explanations of the menu items present in the tool, is in Appendix 2. This user manual can be also be accessed from the login page of the tool. A tutorial appears in Appendix 3; this tutorial can also be accessed from the login page of the tool. Finally, Appendix 4 contains a database diagram, pictorially depicting the tables and the relationship between the tables used in the Bryce tool. Appendix 5 contains the user survey, which was used for Bryce evaluation.

2 Project Management

This chapter discusses the need for project management of software projects and in particular how Extreme Programming projects are managed. It discusses the various tools that are currently used in XP project management, their advantages and drawbacks.

2.1 Introduction to Project Management

Without the right information, you're just another person with an opinion.
— Tracy O'Rourke, CEO of Allen-Bradley[24]

The objectives of software project management are to set and meet achievable commitments regarding cost, schedule, quality, and function delivered—as they apply to individual development or maintenance projects [24]. Effective measurements help software development groups succeed by enabling them to understand their capabilities. Software groups can then develop achievable plans for producing and delivering products and services [24]. Measurements also enable people to detect trends and to anticipate problems, thus providing better control of costs, reducing risks, improving quality, and ensuring that business objectives are achieved [24].

Project management, used successfully for decades in the construction industry, is still an evolving discipline in information technology [10]. Project management is gaining traction in IT organizations and the results are encouraging [10]. Failure rates are down, costs are down, and success rates are

up [10]. This growth of the project management field has been accompanied by the emergence of planning, estimating, scheduling, and tracking tools designed to support the software process [10]. Project management tools support the project management process by providing a vehicle for planning, executing, and controlling the software development. For example, a project management tool may aid in the creation of a project plan for delivering a certain number of features in a release, assigning tasks to team members for the execution of the plan, and controlling the plan execution by monitoring the progress. The project management tool does not actually manage the project; the project manager manages the project by utilizing the tool [10].

2.2 XP Project Management

Metrics are the basic tool for XP project management [12]. For example, the project velocity metric is the basic measure required for running the Planning Game [12]. The project velocity (or just velocity) is a measure of how much work is being completed on a project [25]. It is calculated by adding the time estimates of all the completed user stories of an iteration [25] that were actually completed in iteration. For example, if ten user stories with an estimated completion time of ten hours each are completed in an iteration, the project velocity is 100.

XP does not suggest managing projects “by the numbers” [12]. Instead, XP uses numbers as a way of gently and non-coercively communicating the need for change [12]. Beck also advocates limiting the number of metrics [12]. His rule of thumb is that three or four measures are typically what a team can

stand at any one time [12]. If there is a shortage of tests being written, Beck suggests that a manager should put up a “Big Visible Chart” [12] with the number of tests written and update it everyday [12]. In this way the manager can communicate the need for improvement.

In XP, a person is assigned the role of a tracker to gather project metrics [12]. Tracking should happen without much overhead [12]. It is the responsibility of the tracker to make the minimum amount of measurement and still be effective [12]. Beck suggests gathering project data, such as development time, from each team member approximately twice each week [12]. The XP manager is involved in XP project management because he/she runs the Planning Game, ensures that metrics are seen by those whose work are being measured, and occasionally resolves issues in which team consensus cannot be achieved [12].

2.3 Tools for XP project management

In XP, user stories replace requirements documents [26]. User stories are short, natural-language requirements written by the customers on index cards. User stories are generally one or two sentences and provide just enough detail to make a reasonably low-risk estimate of how long the story will take to implement. The user story card is a commitment between the customer and the developer for a future conversation to clarify the requirement [26]. Before and during the implementation of the story, the developers discuss the details of the story with the customer [26].

After the user story cards are written, customers place priorities on the cards, and developers place time estimates on the cards. If the time estimate indicates a certain story is too large for one iteration, the customer is asked to split the requirement into multiple user story cards. When all priorities and estimates have been completed, the developers and the customer play the Planning Game and collectively decide what features will go into a release. The developers and customers collectively understand the amount of time and resources available for an iteration. They lay the user story cards out on a table and slide them around until they jointly choose the highest priority (and/or highest risk) stories that fit into the iteration.

Once the stories are chosen for a release, the developers choose stories to be responsible for in the release. The story owners then break the stories into smaller tasks (generally a half to a full day) and record this information on task cards. The story owner finds a partner for each task and incrementally writes test cases and implementation code for the task. When the pair completes the implementation of a task and all the new and existing test cases are passed, they integrate and release the new code to the code base and project test suite [12].

The index cards upon which the stories are written have an advantage over computerized media in that they can be organized on a conference room table [27]. But the disadvantage is that they can be at only one place at one time. Sometimes the teams or the customer may be in different locations and

accessing the story or task information can only be possible with some form of electronic media.

XP also calls for some quantitative record keeping and computation of statistics; for example, estimated and remaining days of effort on tasks [27]. Someone is usually delegated the role of tracker, collecting the above information on a personal database (for example, a spreadsheet such as Excel) capable of performing a wide variety of computations on a variety of data. Again, there are times when a physically distributed team would benefit from being able to collect measurements online.

To address the issue of accessing project information online, RoleModel Software, Inc., of Fuquay-Varina, North Carolina used a private Wiki site (discussed in Section 2.4.3) [27]. Another tool, named MILOS ASE (where MILOS stands for “Minimally Invasive Long-term Organizational Support” [28] and ASE stands for Agile Software Engineering [28]), discussed in Section 0, has been developed to provide project collaboration, coordination and organizational learning for distributed XP teams [28].

2.4 Collaboration and Discussion Tools

Collaboration and discussion tools are software that promote and mediate discussions and enable joint working between different users [27]. There are three models for collaboration over a network [27]

- *Email (including mailing list).* Email exchange provides direct exchanges between the members of a collaborative group. The

public form of email is the Internet newsgroup, where postings are distributed globally to all participating newsgroup members and kept there for later access. The disadvantage of postings is that they cannot be edited or easily cross-linked in anyway useful to a group unless they are somehow collected with annotation in a central archive. If archival access is required, postings can made directly to a database, similar to the other models.

- *Shared folder/ File Access.* Shared file access in its simplest form allows members to directly access the same files in a common repository on a public server. The disadvantage of this model is that the files may be in application specific formats and therefore may not be immediately “browsable”.
- *Interactive content update/access.* Interactive page access allows members of a group to collectively edit a material. There are a number of solutions supporting this model, some of the popular ones being conferencing, whiteboard collaboration, etc. Wiki also belongs to this model of collaboration. These interactive update/access models will be discussed in detail in the next few sections.

2.4.1 Conferencing

Computer-based conferencing [29] is an increasingly popular business application that enables distributed or virtual businesses to operate when the

decision makers are geographically separated. Some of the different types of conferencing are [29]:

- *Text-Based Conferencing.* Text-based conferencing is usually asynchronous communication when used in the business world. It is based on the concept of a discussion thread where different participants discuss a particular topic. A thread consists of a group of postings from different participants indented to show replies to an original comment. Synchronous, text-based conferencing is possible using chat software like Internet relay chat.
- *Video Conferencing.* In the last few years, the Internet has accelerated the growth of video conferencing. The advantage of video conferencing is that it reduces travel time and costs and offers an opportunity to arrange a meeting at relatively short notice.

2.4.2 Whiteboarding

Whiteboard software [29] is electronic meeting software. In whiteboarding software, a shared figure is drawn using a stylus, and this annotation is replicated to all others viewing this virtual whiteboard. White boarding software typically provides facilities like zooming, typing text, drawing lines, importing images and erasing using a mouse or a pen type of pointer. White boarding software is often used for brainstorming and summarizing decisions in a project.

2.4.3 Wiki

The Wiki [27, 30], a general purpose collaborative webpage development tool, is server software originally authored by Ward Cunningham. The Wiki allows users to freely create and edit webpage content using any forms-enabled web browser. Wiki supports hyperlinks and has simple text syntax for dynamically creating new pages and cross-links between internal pages. It is a medium for collaboration because of its freedom, ease of access and use, simple and uniform navigation conventions, and apparent lack of formal structure [27]. “Wiki is inherently democratic” [27] – every user has exactly the same capabilities/authority as any other user. It allows web collaboration without the need to manage accounts and passwords for server access; however Wiki can run on intranets and can sometimes have password access to the webpages. Although this may seem risky, experience shows that little damage is done to Wiki content even in the absence of security mechanisms [27].

The following case study illustrates how a Wiki was used in a particular XP project. The XP Wiki system described is a subscription service of Cunningham & Cunningham, Inc. in Portland, Oregon. RoleModel Software, Inc., of Fuquay-Varina, North Carolina, used the service to coordinate an Extreme Programming project for a client in Durham, North Carolina. A separate Wiki web page was created for each story and task of an XP project. Pages were also created for each iteration and for each developer. The iteration pages listed the stories and tasks currently assigned to each iteration. Similarly, each developer page listed

the stories and tasks for which a team member had accepted responsibility for the current iteration.

If a task was delayed to a later iteration, the citation was simply removed from one iteration page on Wiki and added to a later one. Anyone could make such a change simply by editing both iteration pages, cutting the citation from one page and pasting it into the other. Reassigning a task from one individual to another was handled in the same manner as moving task between iterations.

Within the context of the Wiki, customizable calculations could be invoked on recorded story and task pages using CGI scripts. Locating the data within the page to be used as input to the calculation was typically based on a formatting convention. The page formatting convention used to specify the location of the user story estimates in a page is “space indented, one or more words followed by a colon, then (with or without spaces) numbers” [27]. A regular expression is used to parse the page according to this formatting convention and to obtain the user story estimates. Additionally, the CGI script located any references or hyperlinks to further pages, which it recursively walked through. The scripts also contained some other constraints for obtaining data, such as which links to follow and how much of the page to continue for parsing data. A report was generated based on the results of the calculations and was displayed using HTML unordered lists and HTML tables.

There are some technical drawbacks in using Wiki as in the project above:

- Locating data within a page requires a formatting convention for each page. If the pages are not properly formatted, the script may break or produce incorrect output.
- Since a formatting convention is essential for locating the data, changing the layout of a webpage will require modifying the scripts in order to locate the data in the webpage.
- The project Wiki pages have to be recursively walked through to obtain all the project data required for a calculation. It would be more efficient for a program to retrieve project data if the data is stored in a database.

The following table, Table 2.4.1, compares the features available in Wiki and Bryce

Features	Wiki	Bryce
XP project tracking and metrics computation	Requires customization	Yes
Story rework tracking and support for collaboration	Requires customization	Yes
Remote accessibility	Yes	Yes
Built-in support for XP research	No	Some support
Open source	Yes	Yes

Table 2.4.1 Comparison of Wiki and Bryce Features

2.5 Generic Project Management / Workflow Management Tools

There are generic commercial project management tools available; one of the most popular ones is Microsoft Project. The disadvantage of using such a project management tool is that they may require customization for being used in XP projects and also may not aid in XP research. Moreover, the learning curve

for these tools may be too long. Due to the XP culture, programmers would not spend the time to customize or learn the tool and would resort back to manual means of project management.

Another set of tools, called workflow management tools, are used to manage projects. Workflow is defined as a series of tasks within an organization to produce a final outcome [31]. At each stage in the workflow, one individual or group is responsible for a specific task [31]. Once the task is complete, the workflow software ensures that the individuals responsible for the next task are notified and receive the data they need to execute their stage of the process [31]. Generic Workflow management systems share the same disadvantages of a generic project management tools in a XP Project. They may require customization for being used in XP projects, may not aid in XP research and the learning curve for these tools may be too long.

2.5.1 MILOS ASE

MILOS ASE [28] (or short: M-ASE) is a web-based tool that supports agile software engineering practices for distributed teams. It is currently being developed by Dr. Frank Maurer and his team at the University of Calgary. The overall goal of the tool is to support project collaboration, coordination, and organizational learning for virtual software development teams [28].

The main features of M-ASE are listed below [32]:

- *User story management*. M-ASE can track the status of user story implementation in the iterations

- *Iteration planning*. User stories and other tasks are assigned to iterations. The effort available for the next iteration is determined based on work hours and the velocity of the last iteration.
- *Progress tracking*. The system tracks completed user stories and task effort.
- *Metrics collection*. M-ASE manages task effort, task estimates, and estimation accuracy, user stories completed, velocity, source code size metrics, and test metrics.

Both Bryce and M-ASE share many common features. However, they are also different in several aspects.

The following aspects are strengths of M-ASE for use with XP projects:

- M-ASE is integrated with NetMeeting. This allows a programmer to see and converse with another programmer while performing a task even though they may not be collocated.
- M-ASE shows the users who are online at the moment.
- M-ASE can calculate source code metrics such as number of packages, classes, functions and inner classes using JavaNCSS [33]. It can calculate test metrics like the number of JUnit assertXXX statements if the source code is Java.
- M-ASE has a file repository for storing files corresponding to particular task.

Correspondingly, the following are strengths of Bryce for use with XP projects:

- M-ASE support for measuring pair programming is still under development. Bryce can record pair programming information and also compute metrics for pair programming such as amount and percentage of pair programming effort.
- Bryce also enables the quantitative analysis of metrics for research purposes, for example the cost of change/rework of a user story. M-ASE cannot track changes made to a story (because of bugs, customer etc). This information is valuable as it shows how much of rework has been done to a particular story or in a particular release/iteration.

The following table, Table 2.4.1, compares the features in Bryce and M-ASE.

Features	M-ASE	Bryce
XP project tracking and metrics computation	Yes	Yes
Story rework tracking and support for collaboration	No	Yes
Remote accessibility	Yes	Yes
Support for XP research	No	Some support
Integration with NetMeeting	Yes	No
File repository	Yes	No
Computation for source and test code metrics	Yes, if the code is in Java	No
Open source	Yes	Yes

Table 2.5.1 Comparison of M-ASE and Bryce features

3 Introduction to Bryce

This chapter provides an overview of the Bryce tool and its associated documentation. It specifies the tool's development platform and how the tool can be accessed. This chapter also discusses the importance of the tool development for project management and for Extreme Programming research. It provides insight into some important issues like security and portability of Bryce.

3.1 Bryce Overview

Bryce is a tool specially developed for recording and analyzing the metrics of XP, for facilitating project management and reporting on project status and progress. Bryce was developed based on the XP philosophy of "doing the simplest thing that could possibly work" [12] for tracking the project. Hence an important requirement of the tool was that it needed to be easy to use and learn.

In today's fast paced environment, the product requirements change often [10]. Bryce provides support for tracking such changing requirements. Bryce tracks changes made to a user story due to customer requests and also due to other factors like refactoring and bugs fixes.

An important consideration for Bryce's development was the chosen precision of the measurements. The tool does not capture the time spent for the individual phases of software engineering like the design, coding, and testing. This would make the time recording a laborious activity, too laborious for an XP team. More importantly, XP developers iterate through these various

development phases on a minute-by-minute basis, so tracking would likely be impossible. Instead the tool captures the total time for the above phases as task time. Though the time recorded by the tool may lack some precision, it is still very useful. XP researcher Succi states,

> *You will never obtain 100% precision in your measurements, nor do you want it - it is too expensive, and you do not need it. Think of the speedometer-it always shows a 10% error. Still you can use it to drive safely and avoid tickets*. [34]

3.2 List of features in Bryce

The following are the list of features available in Bryce.

- *Automated project tracking*. Project reports can be generated for any release or iteration. The project report provides real-time information regarding the various aspects of the project, such as user story and user task, pair programming, acceptance test, and user story rework information. This information can provide insight into project progress and a manager can take appropriate steps, if necessary.
- *Personal task management*. Team members can track user stories and tasks for which they are responsible in single or multiple projects. The menu item “My Stuff” is specifically dedicated for tracking a team member’s user stories and tasks.
- *Remote accessibility.* Bryce users can easily access and update the project information from anywhere on the Internet (or intranet) with a web browser.

- *Rework tracking.* Bryce users can track rework due to bug-fix, refactoring, pre-release and post-release rework done to a particular user story.
- *Support for collaboration.* Bryce users can record and obtain quantitative information on pair programming.
- *Support for research.* Bryce users can obtain a wealth of quantitative information regarding the project such as pair programming information, rework information on bug-fix, refactoring, pre-release and post-release rework.
- *Support for multiple managers.* Bryce allows more than one team member to perform operations that are privileged to a manager.
- *Documentation.* Bryce has comprehensive documentation available for installing and using the tool. A tutorial is also available for obtaining hands-on experience with the tool.
- *Open Source.* The source code for the Bryce tool is available for customization.

3.3 Bryce Documentation

Three documents have been written for installing and using Bryce. These documents have been included as appendices in this thesis.

- *Installation manual*. This manual explains how to install Bryce on a computer. Please refer to Appendix 1: Installation Manual for more details. The installation manual is based on the initial set up at North Carolina

State University. It was updated when the tool was installed at IBM and Genesys.

- *User manual*. This manual explains how to use Bryce. It contains a list of all the menu items and description of their functionality. A user can access this manual from the login page of the Bryce tool. Please refer to APPENDIX 2: User Manual for more details.
- *Tutorial*. The tutorial guides a new user through the Bryce functionality by giving an example of a pseudo project. A user can access this tutorial from the login page of Bryce. Please refer to APPENDIX 3: Tutorial for more details.

3.4 Platform for Bryce Development and Access

Bryce was developed using JSP™ for server side programming. Being a part of the Java™ family, JSP technology enables rapid development of web-based applications that are platform independent [35]. JavaScript is used to perform client-side validations. Additionally, Cascading Style Sheets (CSS) are used to format the web pages. CSS are used to control style and layout of a webpage and to build a style template that can be repeatedly used in a website [36]. This template approach to designing a site enables developers to create a consistent look for corporate documents, while keeping style separate from content. Bryce uses a single CSS, which contains all the information regarding the style of all its webpages. Bryce requires a Database Management System

(DBMS) for storing project information, a web server with a JSP engine to host the JSP pages, and JDBC™ drivers for database connectivity.

Bryce was developed on a Microsoft© Windows 2000, Internet Information Services 5.0 (IIS) server. The JSP engine, used to serve the JSP pages, was Resin® [37]. Bryce was developed using Microsoft© SQL 2000 as its database server. Other database management systems that support Structured Query Language (SQL), with some advanced features like "subSELECT" (nested SQL) can be used in place of MS SQL server. The aveconnect Version 2.3, Type 4, JDBC™ Driver was used to connect to the Microsoft SQL Server database.

This implementation of Bryce has more than 70 JSP and Java files and over 12000 lines of code. Additionally, it uses 12 tables in two databases to store project information. As mentioned earlier, it uses a single CSS, which contains all the information regarding the style of all its webpages. APPENDIX 4: Database Diagram pictorially depicts the database tables and the relationship between them.

The tool may be accessed using any Web browser at http://bryce.csc.ncsu.edu/xptool/default.jsp and has been tested with Microsoft© Internet Explorer Version 5 and Netscape© Version 6.2.

3.5 Importance of Research

The development of Bryce is important for the following reasons:

- Other available project management tools, like MS Project, may not be able capture information regarding elements unique to XP without some customization; for example, user story information, pair programming information, refactoring information, etc.
- Many project management tools assume project planning is done upfront and hence the entire project plan is generated upfront. This is not suitable for an XP environment where planning is done only for the current release and iteration.
- Some project management tools, like MS Project, cannot be modified or customized to obtain data pertaining to one specific area of research because the source code for the tool may not be available, for example studying the cost of making changes late in an XP Project.

The tool can be utilized to support: 1) project management and 2) research.

3.5.1 Project Management Capability

In XP, user stories replace requirements documents [26]; user stories are still gathered informally on story cards. Subsequently, developers enter user story narratives into Bryce. Developers also enter estimates for completing the user story (in hours) and customer priority for each user story.

When the “View/Edit All Stories” link under the “Feature Management” menu is clicked on the left frame, all the user stories in the current release and iteration can be viewed. Figure 3.5.1 shows a sample list of recorded user stories. This list of user stories is sorted based on the customer priority. Moreover the tasks associated with the user story can be viewed by clicking on the story name. Similarly acceptance tests associated with the user story can also be viewed by clicking the corresponding acceptance test link.

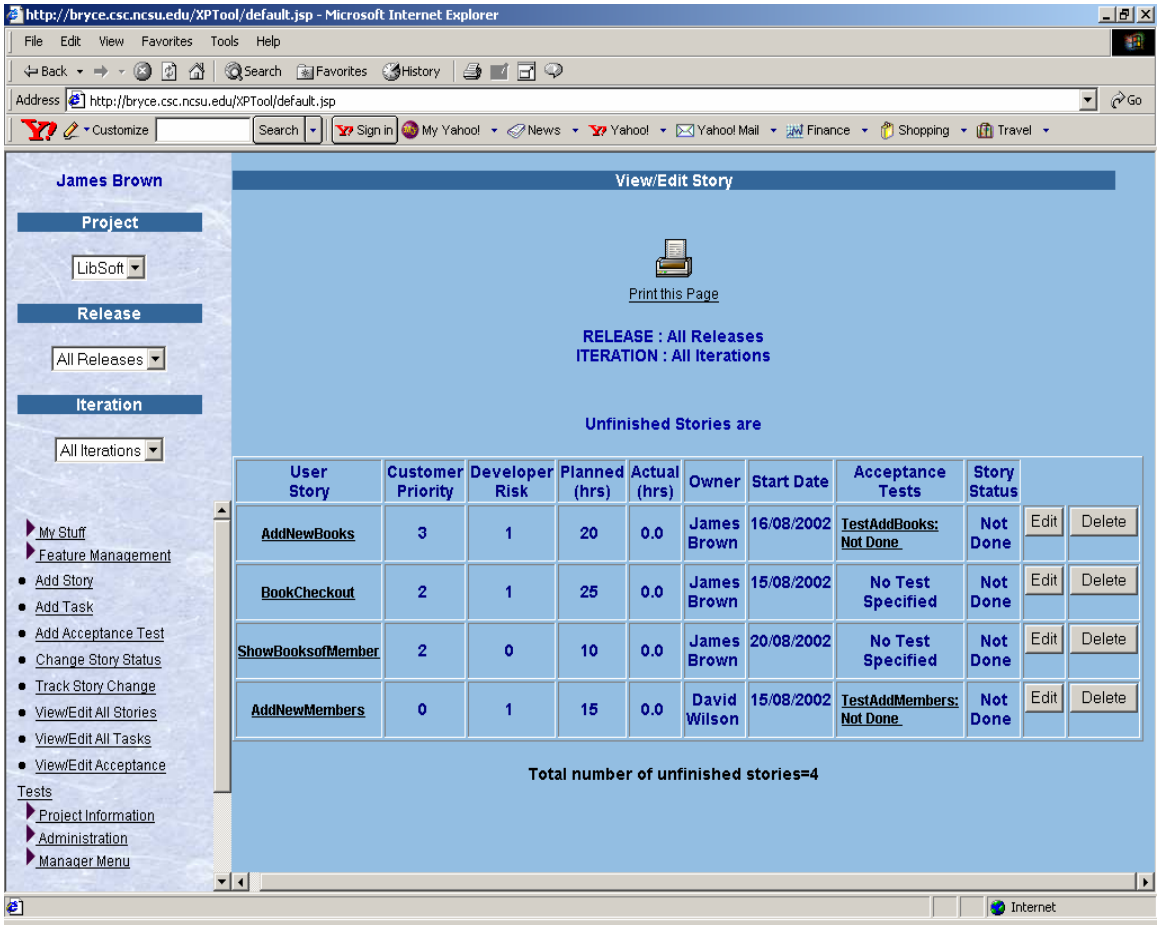

http://bryce.csc.ncsu.edu/XPTool/default.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://bryce.csc.ncsu.edu/XPTool/default.jsp
James Brown
Project
LibSoft
Release
All Releases
Iteration
All Iterations
My Stuff
Feature Management
Add Story
Add Task
Add Acceptance Test
Change Story Status
Track Story Change
View/Edit All Stories
View/Edit All Tasks
View/Edit Acceptance Tests
Project Information
Administration
Manager Menu
View/Edit Story
Print this Page
RELEASE : All Releases
ITERATION : All Iterations
Unfinished Stories are
User Story | Customer Priority | Developer Risk | Planned (hrs) | Actual (hrs) | Owner | Start Date | Acceptance Tests | Story Status
AddNewBooks | 3 | 1 | 20 | 0.0 | James Brown | 16/08/2002 | TestAddBooks: Not Done | Not Done | Edit | Delete
BookCheckout | 2 | 1 | 25 | 0.0 | James Brown | 15/08/2002 | No Test Specified | Not Done | Edit | Delete
ShowBooksofMember | 2 | 0 | 10 | 0.0 | James Brown | 20/08/2002 | No Test Specified | Not Done | Edit | Delete
AddNewMembers | 0 | 1 | 15 | 0.0 | David Wilson | 15/08/2002 | TestAddMembers: Not Done | Not Done | Edit | Delete
Total number of unfinished stories=4
Internet

Figure 3.5.1. Viewing all the user stories in a project

If a story is split into one or more tasks, the task information is also entered in the tool and the task is linked to the respective user story. Also, information regarding acceptance tests for each user story is also entered into the tool. All the stories, tasks and acceptance tests can be viewed and edited.

When the “View/Edit All Tasks” link under the “Feature Management” menu is clicked on the left frame, all the user tasks in the current release and iteration can be viewed. This list of user tasks is sorted based on the task name (or task identifier). Figure 3.5.2 shows a sample list of user tasks that have been recorded the project.

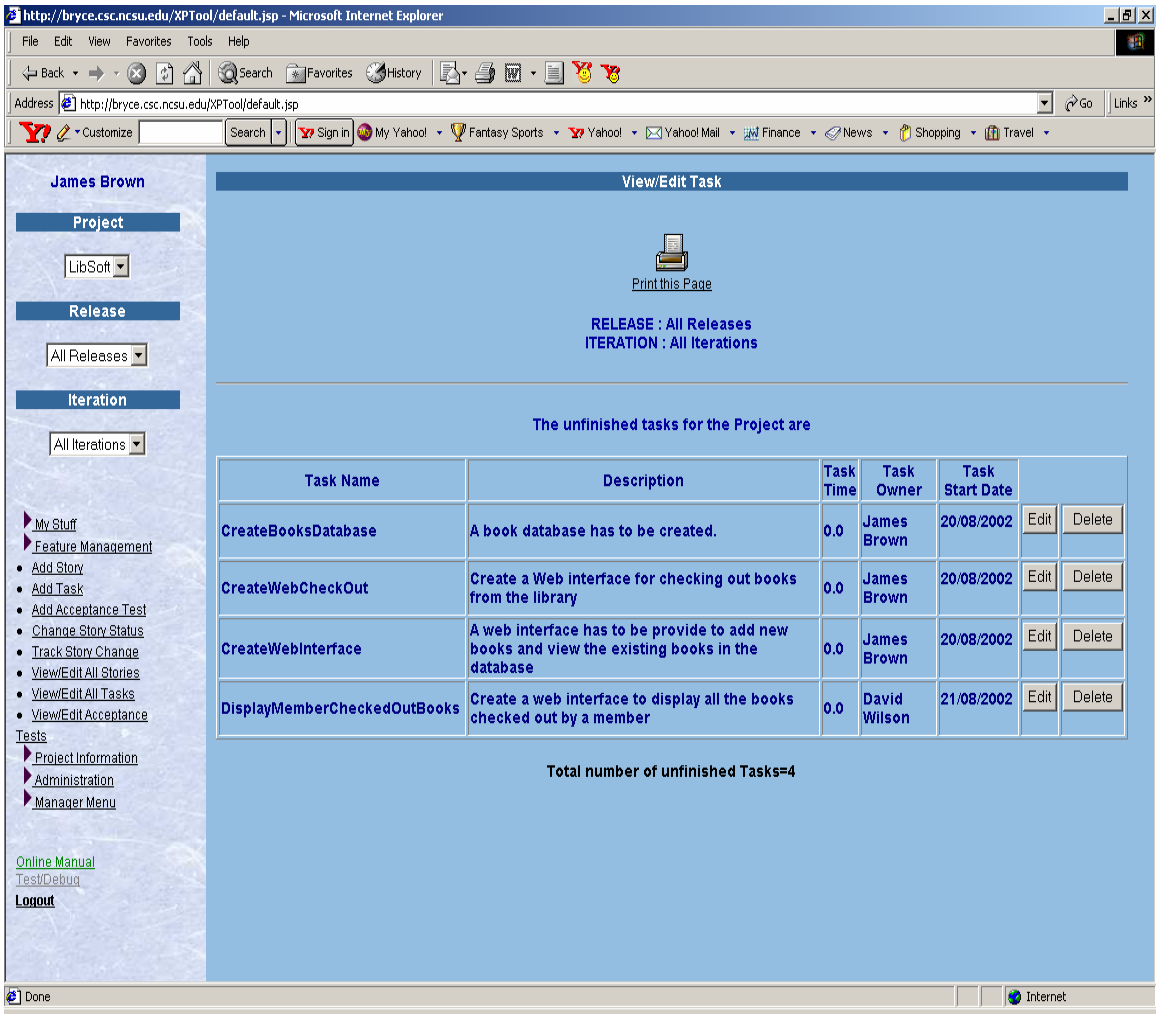


Figure 3.5.2. Viewing all the user tasks in a project

Bryce enables personal task management. The user can view the user stories and user tasks for which he/she is responsible. When the “My Tasks” link under the “My Stuff” menu is clicked on the left frame, all the tasks belonging to the current user be viewed. This task list first displays the incomplete tasks and then the complete tasks. Figure 3.5.3 shows the list of stories for which the current user is responsible.

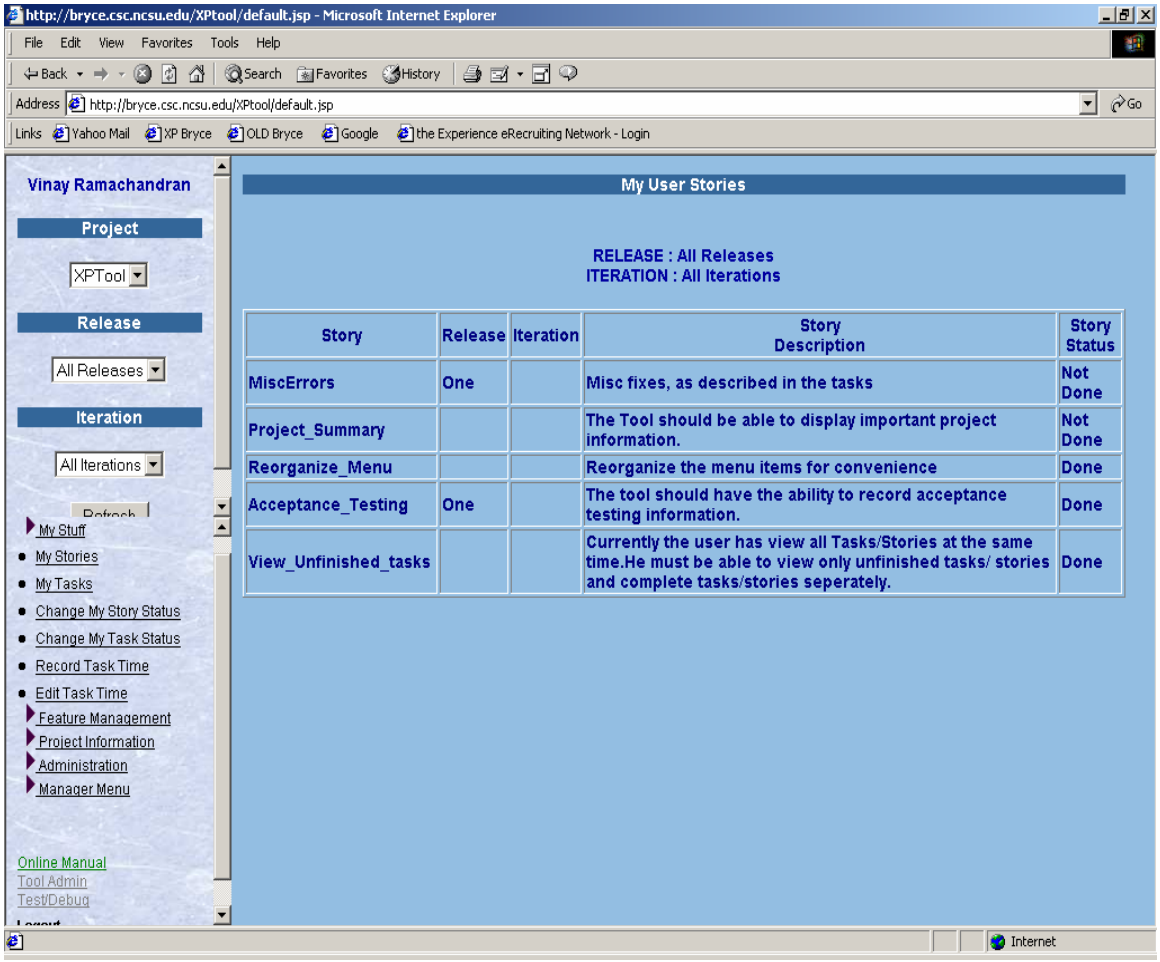

http://bryce.csc.ncsu.edu/XPtool/default.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://bryce.csc.ncsu.edu/XPtool/default.jsp
Links Yahoo Mail XP Bryce OLD Bryce Google the Experience eRecruiting Network - Login
Vinay Ramachandran
Project
XPTool
Release
All Releases
Iteration
All Iterations
My Stuff
My Stories
My Tasks
Change My Story Status
Change My Task Status
Record Task Time
Edit Task Time
Feature Management
Project Information
Administration
Manager Menu
Online Manual
Tool Admin
Test/Debug
My User Stories
RELEASE : All Releases
ITERATION : All Iterations
Story | Release | Iteration | Story Description | Story Status
MiscErrors | One | | Misc fixes, as described in the tasks | Not Done
Project_Summary | | | The Tool should be able to display important project information. | Not Done
Reorganize_Menu | | | Reorganize the menu items for convenience | Done
Acceptance_Testing | One | | The tool should have the ability to record acceptance testing information. | Done
View_Unfinished_tasks | | | Currently the user has view all Tasks/Stories at the same time.He must be able to view only unfinished tasks/ stories and complete tasks/stories seperately. | Done
Internet

Figure 3.5.3. Listing user stories belonging to the current user

The tool also tracks changes made to a user story. When the “Track Story Change” link under the “Feature Management” menu is clicked on the left frame, the user can pick a user story and view the changes made to it. Figure 3.5.4 shows the rework done on a user story named “AddNewBooks”. The figure shows a bug fix change that was made to the user story. This page also shows total rework effort for the user story and percentage of rework effort.

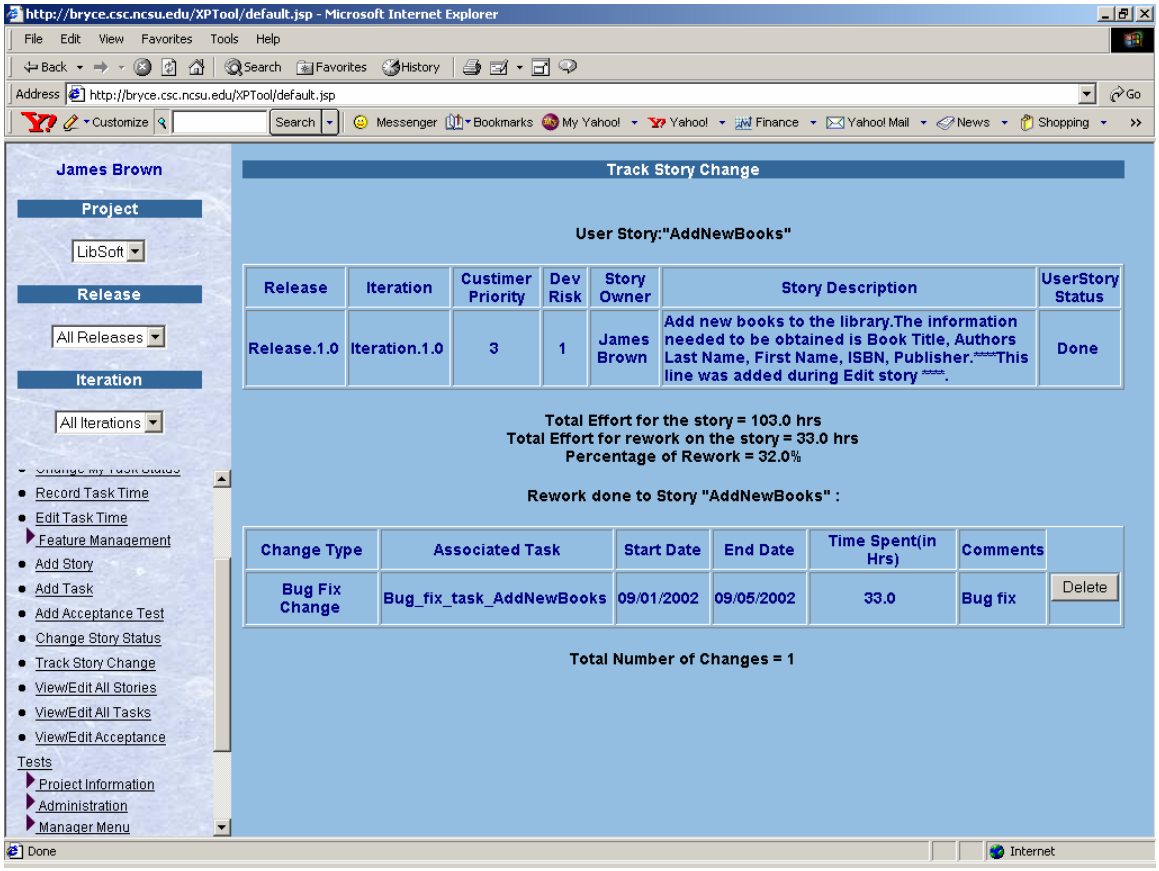


Figure 3.5.4. Tracking the rework done to a user story

Project reports can generated for any particular release/iteration or the project as a whole. The project report helps in tracking project progress and obtaining project metrics real time from anywhere with just a web browser. Figure 3.5.5 shows a sample project report that was generated by the tool. The report contains user story information, user task information, acceptance test information, pair programming information, bug-fix rework information, refactoring rework information, pre-release rework information, post-release rework information and overall rework information.

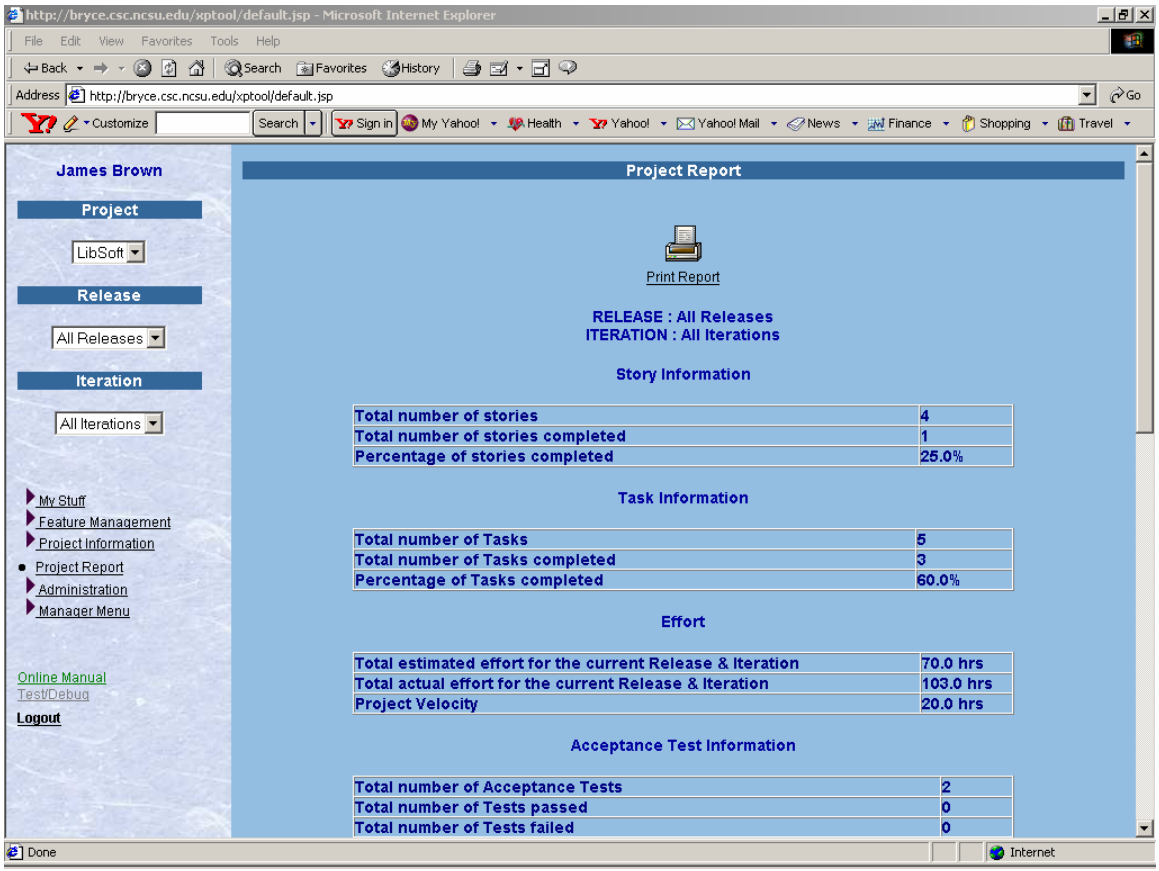


Figure 3.5.5. A sample Project Report

3.5.2 Research on XP

There exists little statistical evidence to prove the effectiveness of XP. To date the evidence is highly anecdotal, but has aroused the curiosity of many highly respected software engineering researchers and consultants [21]. Bryce may aid in obtaining statistical evidence for (or against) the effectiveness of XP projects. Bryce has the capability to generate comprehensive reports for every project release and iteration. These reports provide quantitative feedback regarding a project, which will aid in XP research.

One particular area where Bryce could aid in research is studying the "cost-of-change" curve for XP, shown in Figure 3.5.7. In [38], Boehm analyzes the relative cost of correcting software errors (or making other software changes) as a function of the phase in which the corrections or changes are made. He concludes that as the project progress from analysis to design, coding, testing and production, the cost of making changes to software increases [38] as portrayed in Figure 3.5.6. This is the universally-accepted cost-of-change curve where the cost of making changes to software rises exponentially with time.

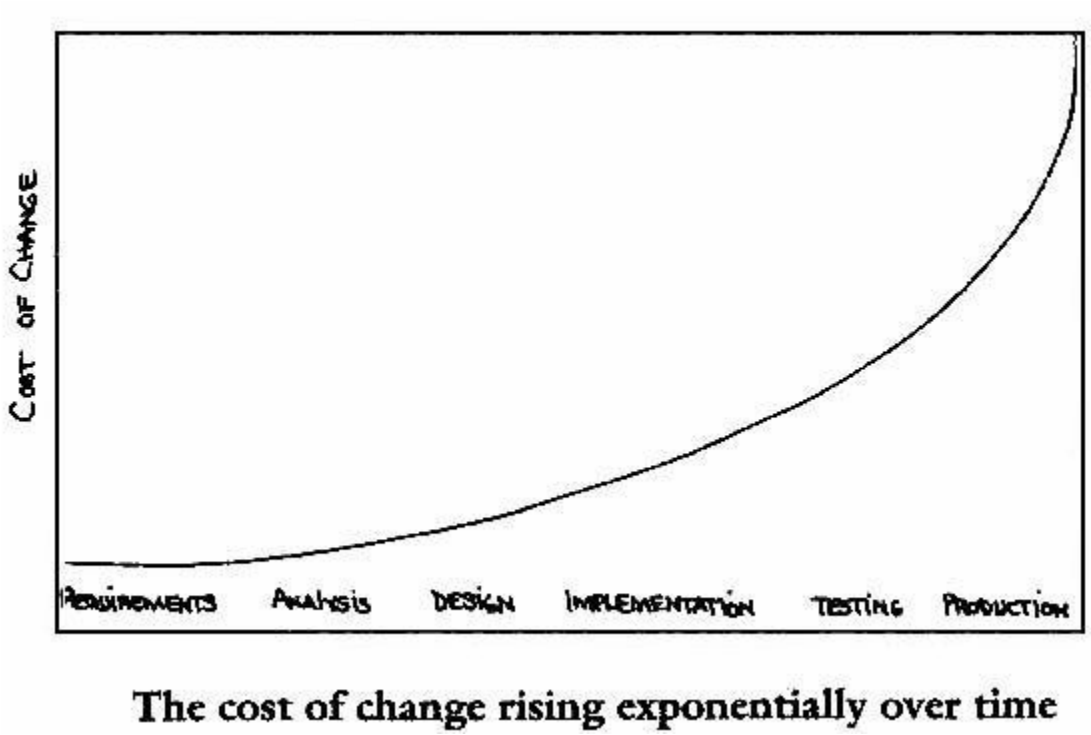


Figure 3.5.6 The Cost of Change Curve for Traditional Processes (source [12])

A flattened cost of change curve is "*the* technical premise of XP" [12]. In his book [12], Kent Beck states, "Under certain circumstances, the exponential rise in the cost of changing software over time can be flattened". This is portrayed in Figure 3.5.7.

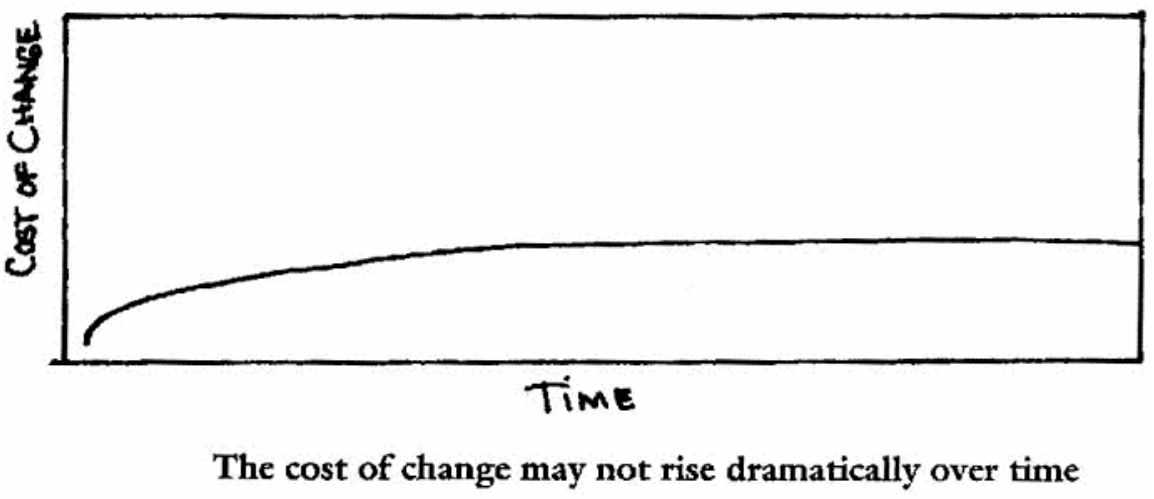


Figure 3.5.7 The Cost of Change Curve for XP (source [12])

However, there is no quantitative evidence to support this claim. A flattened cost-of-change curve allows big decisions to be made as late in the process as possible, defers the cost of making the decisions and have the greatest possible chance of being right [12]. Research has shown that developers perceive that XP allows change in software without incurring big costs [23], though there is no quantitative evidence to prove the same.

In order to ascertain the flat cost of change, Bryce can be utilized to gather data from real XP projects. Bryce can compute and display the cost of change information for every release/iteration, and this can be viewed in the Project Report (see Figure 3.5.8) generated by the tool. The report contains information on each of the different types of rework i.e. bug-fix, refactoring, pre-release, post-release and their aggregate rework information. Bryce can compute the amount of rework (in hrs), effort for rework (in hrs), percentage of rework effort and

average cost per rework for each of the different types of rework (in hrs) and also their aggregate rework for every project release and iteration. Figure 3.5.8 shows the rework information of a sample project. This information may be used to plot and quantitatively ascertain whether the cost of change curve is somewhat flat for XP.

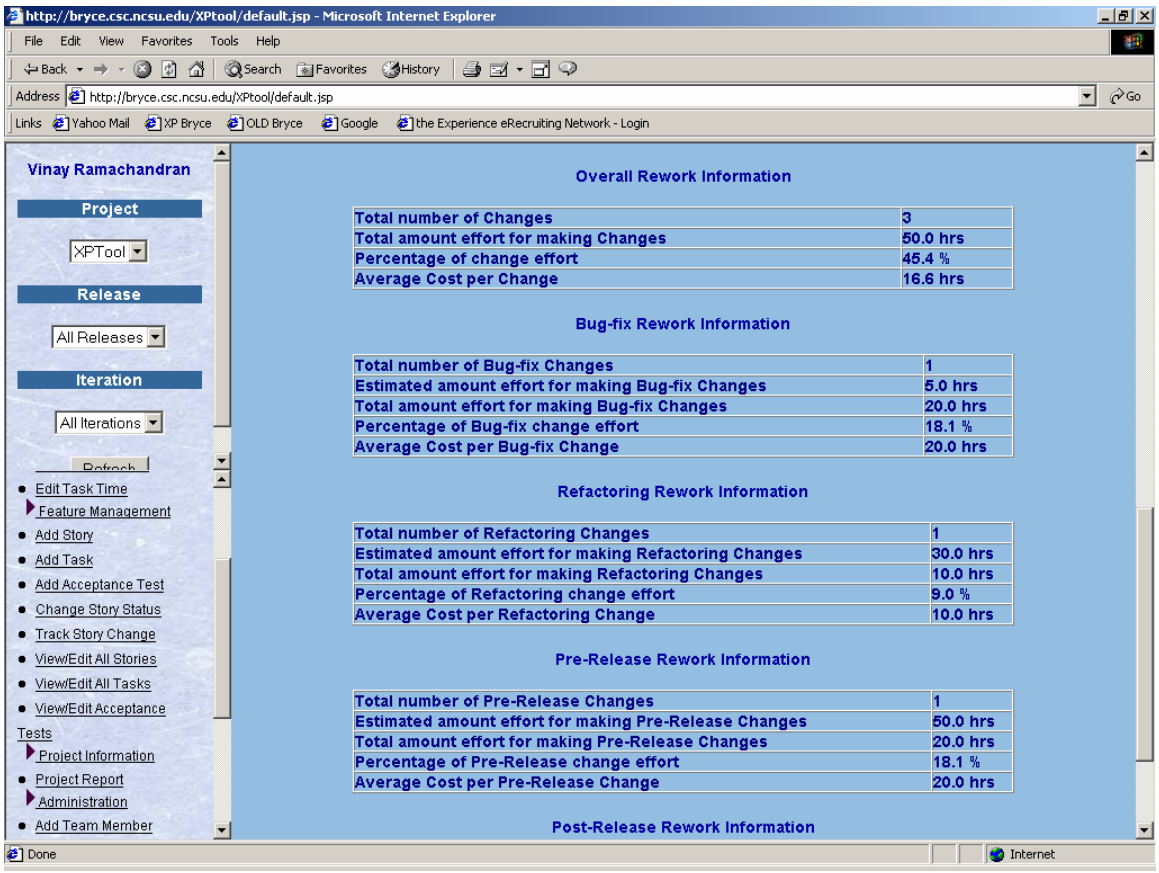

http://bryce.csc.ncsu.edu/XPtool/default.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://bryce.csc.ncsu.edu/XPtool/default.jsp
Links Yahoo Mail XP Bryce OLD Bryce Google the Experience eRecruiting Network - Login
Vinay Ramachandran
Project
XPTool
Release
All Releases
Iteration
All Iterations
Refresh
Edit Task Time
Feature Management
Add Story
Add Task
Add Acceptance Test
Change Story Status
Track Story Change
View/Edit All Stories
View/Edit All Tasks
View/Edit Acceptance Tests
Project Information
Project Report
Administration
Add Team Member
Overall Rework Information
Total number of Changes 3
Total amount effort for making Changes 50.0 hrs
Percentage of change effort 45.4 %
Average Cost per Change 16.6 hrs
Bug-fix Rework Information
Total number of Bug-fix Changes 1
Estimated amount effort for making Bug-fix Changes 5.0 hrs
Total amount effort for making Bug-fix Changes 20.0 hrs
Percentage of Bug-fix change effort 18.1 %
Average Cost per Bug-fix Change 20.0 hrs
Refactoring Rework Information
Total number of Refactoring Changes 1
Estimated amount effort for making Refactoring Changes 30.0 hrs
Total amount effort for making Refactoring Changes 10.0 hrs
Percentage of Refactoring change effort 9.0 %
Average Cost per Refactoring Change 10.0 hrs
Pre-Release Rework Information
Total number of Pre-Release Changes 1
Estimated amount effort for making Pre-Release Changes 50.0 hrs
Total amount effort for making Pre-Release Changes 20.0 hrs
Percentage of Pre-Release change effort 18.1 %
Average Cost per Pre-Release Change 20.0 hrs
Post-Release Rework Information
Done
Internet

Figure 3.5.8. Rework information of a project

Bryce may also be used to access the effectiveness of XP. Williams, et. al. [39] suggest a metric suite for evaluating an agile methodology. These metrics, which will be discussed shortly, obtained from an XP project have to be compared with a traditional process to evaluate the effectiveness of XP. Bryce

can aid in collection of some of the metrics suggested in [39]. These metrics are obtained from the “Project Report,” which are can be generated by Bryce:

- *Productivity and Cycle Time.* The developer productivity is the amount of time required to complete a task in relation to a measure size, such as lines of code or function points [39]. Bryce computes the average time taken to complete a user story in every release or iteration. It also computes the effort required for completing tasks, releases and projects.
- *Responsiveness to Customer Changes.* XP professes the ability to adapt to customer changes during product development [12]. The customers may make Service Requests (SRs) to the development team to change the existing functionality because of defects or desire for enhanced functionality [39]. The ability to adapt to customer change could be assessed by counting the number of SRs that deal with altering the previously released functionality [39] and computing the amount of time spent by the development team changing previously released functionality as a percentage of total development effort [39]. Bryce automatically tracks the number of post-release rework performed for user stories. It also computes the amount of post-release rework effort and the percentage of post-release rework effort to the total development effort in a release or iteration. Additionally, it also computes the average effort per user story required for post-release. The above information is also

available for the other types of rework, such as bug-fix, refactoring, pre-release rework.

3.6 Benefits of using Bryce

The following are the benefits of using Bryce in an XP project

- Bryce specifically is developed for XP. Hence it can capture information regarding elements unique to XP such as user story, refactoring, and pair programming information.
- It provides real-time information about a project, such as project velocity, user stories, tasks, acceptance tests, and pair programming information etc.
- It is web-based and, hence, can be accessed anywhere on the Internet or via an intranet.
- It has a low learning curve and little training is necessary to learn the tool. Ninety three percent of the students in CSC517, ECE 521/ECE 463 and CSC326 took just a few hours to learn the tool.
- It lessens the burden or even eliminates the need of a “tracker” [12]. Once the data is entered into the tool by each developer, the tool tracks the project.
- Since the source code is available, new features can be added according to the new user requirements and the research area of interest.

Moreover the tool is designed to be scalable because more features can be added with minimal or no changes to the existing functionalities.

3.7 Portability

Bryce is designed in such a way that it can be ported from one system to another in future. Broadly, the Bryce can be divided into following components,

- JSP, Java files and HTML files
- Database management system (DBMS)
- JDBC Drivers to connect to the database server

As mentioned before, Bryce was developed on a Microsoft© Windows 2000, Internet Information Services 5.0 (IIS) server. The JSP engine, used to serve the JSP pages, was Resin® [37]. Bryce was developed using Microsoft© SQL 2000 as its database server. The aveConnect version 2.3, type 4, JDBC™ Driver was used to connect to the Microsoft SQL Server database.

In order for Bryce to be portable, each of the above components has to be portable. Java [40] files, JSP [35] and HTML files [41] are platform independent and can be ported easily to another platform. Hence they can be ported to another web server using a JSP engine on a different platform. Other database management systems that support Structured Query Language (SQL), with some advanced features like “subSELECT” (nested SQL), can replace MS SQL as the DBMS. Since MySQL© 4.0 does not support the “subSELECT” feature [42], it cannot be used as DBMS for Bryce. In addition, the Bryce’s databases would

need to be migrated to the new DBMS. This can be done easily using commercial tools such as SQLPorter©. However, SQLPorter© does not support migration to many DBMS as yet. Finally, JDBC drivers are required to connect to the DBMS, which preferably should be Sun certified [43]. The Sun JDBC certification helps product vendors and applications developers choose the appropriate JDBC drivers for their specific needs with confidence that the driver has been certified to work with Java 2 Platform, Enterprise Edition (J2EE) compatible products [43].

At IBM and Genesys Telecommunications Laboratories, a different web server and JSP engine were used than the setup at North Carolina State University. In these locations, Apache version 1.3 was used instead of Microsoft© Internet Information Services 5.0 (IIS) as the web server, and Tomcat version 4.x was used in place of Resin® as the JSP engine. Since Bryce has not been used very much at these industrial locations, we are not sure which configuration (IIS & Resin or Apache & Tomcat) is better.

3.8 Security

Bryce has security features to prevent unauthorized access of project data. Bryce has two kinds of users, the normal user and the manager. The manager can perform some additional operations (discussed in Section 4.4.5) than a normal user. Bryce uses session objects to share information across its JSP pages. Once a user successfully logs into the system, a session object is created. This session object contains information regarding the user, including

their user Id. Invalid access to a webpage is denied by first checking whether the session is a valid session and the user is a valid user. For example, the user may try to access a webpage directly by typing the URL without actually logging into the system. In this case, the session is invalid and consequently an “Invalid Session” page error is returned. Similarly invalid access to a restricted manager’s page is denied by verifying whether a user is a valid manager. To perform the above validations a Java API called “UserValidation” was implemented. This API is used in JSP to verify whether a user is indeed a valid user/manager.

Collective Ownership is one of XP practices [12]. It implies that anyone can change any code anywhere in the system at anytime, regardless of who originally created the code. Bryce extends this XP philosophy to allow anyone on a team to add, delete or modify project information. However some operations like creating or deleting the projects, releases, iterations are restricted a project’s manager since these operations must be performed by the manager and not by the team members. Restricting access to functions in this way prevents catastrophic operations like deleting a project, which obviously cause all the project information to be deleted. However in the future version of Bryce, it would be useful to have a logging mechanism, which will list the changes that are made to a project.

However Bryce does possess some security limitations. Bryce stores passwords in a database and does not encrypt them. Nor is data encrypted when sent across the network. This is not critical in an academic setting as project

confidentiality is not critical. However in a corporate setting, where confidentiality is vital, Bryce must be shielded by using a firewall. Currently, Bryce installed on the Intranet in IBM and Genesys is protected by the company's network firewall. In the future, secure servers, which use Secure Sockets Layer (SSL), should be an alternative solution to firewalls.

4 Bryce Features

This chapter discusses the functionality present in Bryce. It explains how a new user can sign up with the tool, set up a new project and utilize the tool for project management. This chapter also provides explanation to all the menu items present in Bryce.

4.1 Bryce Features Overview

Bryce is currently accessible at the following URL http://bryce.csc.ncsu.edu/XPTool/default.jsp. A user must first register to the tool using the "New user sign up" link in the login page. Once a new user has registered, he/she will be able to login to the tool and use it. Bryce has three categories of users

- *Normal user* - A normal user may use Bryce to record/view project information (except the information discussed in Section 4.4.5). In a project, typically a team member belongs to this category of users. A normal user can:
    - add/edit User Stories, User Tasks, Acceptance Tests pertaining to the Project
    - view/record certain Project related information such as viewing project report
    - perform limited project administration tasks like changing the current user's password

- *Manager* - A manager has certain special privileges in addition to the above listed normal user privileges. Typically, a project manager or team leader is granted the “manager” privileges. In addition to the normal user privileges, the manager may also
    - add/delete projects
    - add/delete release information
    - add/delete iterations of a release
- *Administrator* - The administrator is responsible for the overall too maintenance. The tool administrator may grant or take away manager privileges from a user. The tool administrator is responsible for backing up the database periodically and restoring it in the event of a system crash.

The generic term “user” employed in the remaining sections of this thesis refers to both normal users and managers. Once users are registered as a normal user or manager, they can login and begin using Bryce.

4.2 New User Registration

To use Bryce for the first time, users must register. All users must first register as normal users and may then request the tool administrator to grant manager privileges. As mentioned earlier, only the tool administrator can add or remove managers.

The “New user sign up” link on the login page is used to register new users. The user must specify a user Id (which is also the email address of the user) when registering. Additional information required for registering a user to

the tool are password, first name, last name and the project manager name, which is chosen from a project manager list in a drop down menu. Once users are successfully registered, an email notification is automatically sent to the manager. The manager may then add the new team member to the project. If a user has been successfully registered and the manager has added the user as a project team member, the user may login to access their assigned project information. A sample registration can be viewed in APPENDIX 3: Tutorial.

4.3 Using Bryce

To begin working on a project, the user must first select a project from the project dropdown menu (see page 92). Then, the user can opt to work on a particular release and iteration by selecting them in their corresponding drop down menus (see page 98).

4.4 Description of Bryce menu items

Once a project is selected from the project drop down menu, users may access the project information. The following menu items allow the user to add, delete, view or edit the various aspects of the project information. More details about the menu items can be found in APPENDIX 2: User Manual and APPENDIX 3: Tutorial.

4.4.1 My Stuff

The “My Stuff” menu item provides users with the capability of tracking stories and tasks for which the current user is responsible. The following submenu items are provided:

- **My Stories:** This submenu item is used to view user stories assigned to the current user.
- **My Tasks:** This submenu item is used to view user tasks assigned to the current user.
- **Change My Story Status:** This submenu item is used to change the story status of the current user. The following are the valid story states: “not done,” “done,” “pre-release change,” ”post release change,” “refactoring,” and “bug fix change”. A user story is initially in “not done” state. When a story is completed, it is changed to the “done” state. When a story requires rework, the story state is changed to one of the following rework states: “pre-release change,” ”post release change,” “refactoring,” or “bug fix change” depending on the type of rework. A rework task is also added to the story in order to track the time spent on this story rework. Once this rework is completed, the task status is set to “complete”. Now the story can be changed back to “done” state.
- **Change My Task Status:** This sub menu item is used to change the task status of the current user as completed when the task has been completed.

- **Record Task Time:** This submenu item is used to record time to user tasks assigned to the current user. The user specifies the amount time spent on the particular task and the collaborators involved in the task sessions.
- **Edit Task Time:** This submenu item is used to edit the recorded time spent on tasks of the current user.

4.4.2 Feature Management

This menu item provides the user with the capability of adding, modifying or viewing any story, task or acceptance test in the project. The tool assumes that there is trust among the project team members and hence allows all the team members to view or edit project information.

- **Add Stories:** This submenu item is used to add new user stories to the project. The story can be assigned to a particular release or iteration either while adding a story or sometime later using the “View/Edit Stories” submenu item.
- **Add Task:** This submenu item is used to add new user tasks. The task can be assigned to a particular user story either while adding a task or sometime later using the “View/Edit Tasks” submenu item.
- **Add Acceptance Tests:** This submenu item is used to add new acceptance tests. Each acceptance test is associated with a user story and the associated user story must be specified while adding acceptance tests.

- **Change Story Status:** This submenu item is used to change the story status of any user. The following are the valid story states: “not done,” “done,” “pre-release change,” ”post release change,” “refactoring,” and “bug fix change”. As shown in Figure 4.4.1, the user story is initially in “not done” state. When the story is completed, it is changed to “done” state. When rework is to be done on a story, the story state is changed to one of the following rework states: “pre-release change,” ”post release change,” “refactoring,” or “bug fix change”. A rework task is also added to the story in order to track the time spent on this story rework. Once this rework is completed, the task status is set to completed. Now the story can be changed back to “done” state.

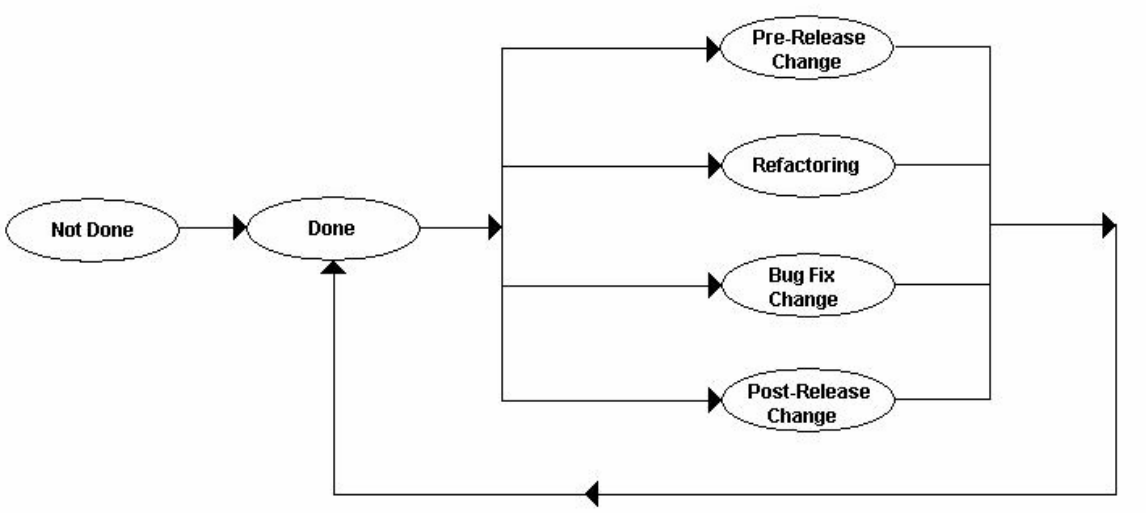


Figure 4.4.1. Story Traversing Different States

- **Track Story Change:** This submenu item is used to view all the rework that has been done to a story. Apart from the description of the story rework, the tool also computes and displays quantitative information regarding the rework. This includes total effort for the story, total effort for rework on the story and percentage of rework for the story.
- **View/Edit Stories:** This submenu item is used to view/edit user stories and tasks. He can view the tasks corresponding to a particular story by clicking on the story identifier. The “Edit” buttons beside the stories and tasks are used to edit the corresponding stories and tasks. Once the “Edit” button is clicked the individual fields of the story or task can be edited. Also, stories and tasks can be deleted clicking the “Delete” button beside them.
- **View/Edit Tasks:** This submenu item is used to view/edit user tasks. On clicking the “Edit” button, the individual fields of the task can be edited. Also, the tasks can be deleted by clicking the “Delete” button beside them.
- **View/Edit Acceptance Tests:** This menu item is used to view/edit acceptance tests (AT). On clicking the “Edit” button, the individual fields of the AT can be edited. Also, the acceptance tests can be deleted clicking the “Delete” button beside them.

4.4.3 Project Information

This menu item is used to obtain information regarding the current project.

- **Project Report:** This menu item is used to compute and display metrics pertaining to the project. If a release and iteration is chosen in the corresponding dropdown menus, the project report will contain the corresponding release and iteration specific information. The project report contains quantitative information regarding the following:
    - Story Information
    - Task Information
    - Acceptance Test Information
    - Estimated and Actual Effort Information
    - Pair Programming Information
    - Story Rework Information

4.4.4 Administration

This menu item is used for the administration of the project. The submenu items are

- **Add Team Member:** This submenu item is used to add a new team member to the project. In order to do so, the new team member must have already registered to the tool. Additionally, he/she must have chosen a project manager for the current project as his/her manager before he/she can be added as a team member to the project. To add

the new team member, the username and the password must be entered after clicking this submenu item.

- **Delete User Registration:** This submenu item is used to delete a user from the tool. This menu item deletes the currently logged user from the system. To delete the user, the user's password must be entered.
- **Add/Change Manager:** This submenu item is used to change the current project manager. The user can choose a new manager from a list of project managers registered to the tool.
- **Change Password:** This submenu item is used to change the password of the current user.

4.4.5 Manager Menu

This menu item is only visible to a Manager. The manager, to perform certain privileged tasks, uses this menu item. The submenu items are:

- **Create Projects:** The manager uses this submenu item to add a new project.
- **Delete Project:** The manager uses this menu item to delete a project from the tool. The manager will be required to confirm the deletion of the project. Once a project has been deleted, the project related information is lost forever and cannot be recovered.
- **Create Release:** The manager uses this submenu item to add a new release to the project.

- **Delete Release:** The manager uses this submenu item to delete a project release. The manager confirmation is required for the deletion of the release.
- **Add Iterations:** The manager uses this submenu item to add iterations to a project release.
- **Delete Iterations:** The manager uses this submenu item to delete iterations of the project. The manager confirmation is required for the deletion of the iteration.
- **Add/Remove Team member:** The manager uses this submenu item to add/delete team member(s) to/from a project. The manager can add a user to the team only if the user has registered to the tool with him as the manager.

4.4.6 Other Menu Items

These menu items are:

- **Online Manual:** This menu item is allows the user to access the online manual for the tool.
- **Tool Admin:** This menu item is used by the tool administrator to add managers to the system
- **Test/Debug:** This menu item is used for debugging purposes.
- **Logout:** This menu item is used to log out of the tool.

5 Preliminary Bryce Evaluation

This chapter explains how Bryce was evaluated using a survey. It discusses the details of the survey and analyses the survey responses. It also discusses the user feedback and how Bryce may be useful in XP research.

5.1 Method for Bryce Evaluation

In section 1.4, we outline the following two hypotheses in relation to this tool:

1. A lightweight project management tool will be beneficial for managing XP projects.
2. A lightweight measurement and analysis system will aid in research by obtaining quantitative information regarding XP projects.

The above hypotheses was evaluated through its use in two graduate-level courses, CSC517 [Object-Oriented Languages and Systems, 49 students] and ECE 521/ECE 463 [Computer Design & Technology / Advanced Microprocessor Systems Design, 49 students], and one undergraduate course CSC326 [Software Engineering, 140 students] at North Carolina State University in the Fall 2002 semester. The students were required to use the tool as a part of their project. Additionally, Bryce was installed for use at two industrial locations: International Business Machines (IBM) and Genesys Telecommunications Laboratories. Feedback was obtained from the users of Bryce using a survey. The details of the survey will now be discussed.

5.2 Details about the Survey

The online survey (see APPENDIX 5: Bryce Survey) had 17 multiple-choice questions with a set of answers to choose from. The possible answers to these questions conformed to Likert scaling to aid in statistical analysis. The answer choices were discrete continuums, such as “Strongly Disagree”, ”Disagree”, “Agree” and “Strongly Agree.” These choices were scaled linearly from 1 to 4. There were no neutral choices like “Neither Agree nor Disagree”. Hence, the choices were such that they ensured that the respondent to express a definite opinion, either positive or negative. A few questions also required the respondent to type in entire qualitative responses, such as suggestions for making the tool easier to use/navigate.

The quantitative data was analyzed statistically, using metrics such as mode and the percentage of the participants choosing the mode of the answers chosen by the participants for a particular question. In many cases, the answers were aggregated to represent the perceptions of a greater percentage of participants. For example, answers like “Agree” and “Strongly Agree” were aggregated as positive opinions. In some cases, the final values were rounded off to the nearest integer since their decimal values were not very significant.

5.3 User Feedback

The students in CSC517, ECE 521/ECE 463 and CSC326 used Bryce to record and track information pertaining to their project/programming assignments. The project/programming assignments were generally of short

duration, typically lasting a month or less. Students worked in teams comprised of two to four students. Since the project/programming assignments were of short duration, the students generally did not perceive the value of recording project information and of using a tool to track their project. Consequently, only 45% felt that the tool could save time and/or effort in tracking the project and only 43% felt that the project report, which is used for obtaining project status, was useful to them. However, 69% of the students still agreed that accessing project information online was useful to them and 69% stated they used the tool to track their project progress at least some of the time. Ninety one percent of the students encountered five bugs or less while using Bryce and 74% stated that they could access the tool most of the time. We attribute the lower accessibility to the fact that the students were accessing the system on the development machine, which was periodically restarted due to the interference of other applications running on the machine. Moreover, the students felt that the tool was easy to learn. Ninety three percent of the students took just a few hours or less to learn the tool. Yet only 60% said that the tutorial and manual were helpful in learning the tool. This suggests that the many students learnt the tool by trial and error and not by going through the manuals. Only 16% of the students stated that just a single team member entered data or monitored the project using the tool. This shows that in most cases, two or more students now shared the role of the XP tracker, which supports our view that Bryce can reduce the burden on the tracker. However, only 47% of the students agreed that Bryce was easy to use.

Hence making the tool easier is a definite area for improvement. Finally, a majority (67%) of the students agreed that the tool is a better alternative to story cards and 67% said that the tool could be useful in industrial projects.

The following additional information could be obtained from the survey. Overall, only 58% percent of the students liked implementing their project using XP. Also, 53% of participants worked in co-located teams and 71% of the team member practiced pair programming. Lastly, 82% of the participants said that they spent 30 minutes or more a week using Bryce.

The survey was not completed by developers at Genesys or IBM. Although Bryce has been installed in these locations, the XP projects at each location are just beginning. However Bill Krebs, the IBM manager, agreed that Bryce is a better alternative to story cards, can save time and/or effort in tracking a project and can be useful in industrial projects.

We hypothesized that a lightweight project management tool will be beneficial for managing XP projects. Based on our findings, we state that Bryce may be useful for managing XP projects, though not popular, in an academic environment. Initial indications support its usefulness in managing industrial XP projects.

5.4 Bryce for XP Research

At the end of the Fall 2002 semester, quantitative data can be obtained about academic XP projects using the “Project Report” as discussed in Section 3.5.2. Since these projects were of short duration, it will not be possible to obtain

the “cost of change” information discussed in Section 3.5.2 pertaining to these projects; this information must be collected over several releases to obtain statistically valid results. However, Bryce can give instructors and researchers information about the effort required to accomplish each of the modules and the whole project/programming assignment and pair programming information. Indeed, two graduate theses will use the data obtained from Bryce during the Fall 2002 semester. One of these theses will examine the potential of pair programming as a vehicle for project management; the other thesis will study the distributed pair programming technique. Additionally, two supported researchers will be investigating the data collected at IBM and Genesys. To summarize, Bryce is currently being used to support four pair programming and/or XP research studies.

Data currently being collected from industrial XP projects at IBM and Genesys will be used for XP research purposes. This data can be obtained at the end of each release/iteration from the project report that is generated by the tool, a sample of which can be seen in Figure 3.5.5 and Figure 3.5.8. The information includes information on project progress, pair programming, and each of the different types of rework i.e. bug-fix, refactoring, pre-release, post-release and total rework information. This information can provide an insight to the effectiveness of XP and also whether the cost of change curve is somewhat flattened for XP. This supports our second hypothesis that Bryce may aid in research on XP.

6 Conclusion

Bryce is a web-based tool, developed for the following purposes:

1. XP project management.
2. Aiding research on XP by providing a vehicle for obtaining quantitative information regarding XP projects.

Bryce was evaluated through its use in two graduate-level courses and one undergraduate course at North Carolina State University in the Fall 2002 semester. The students were required to use the tool as a part of their project. Furthermore, it has been installed for use for managing XP projects at two industrial locations: International Business Machines (IBM) and Genesys Telecommunications Laboratories. Feedback was obtained from the users of Bryce using a survey.

The students who used the tool generally did not perceive the value of recording project information and of using a tool to track their project, since their project/programming assignments were of short duration. However, the many (65%) of the students still agreed that accessing project information online was useful to them. Most of the students took just a few hours or less to learn the tool. However, many students felt that Bryce was difficult to use. Nevertheless, making the tool easier to use could be one area for improvement. Bryce also has been shown to reduce the burden on the XP tracker.

From our research, we state that Bryce appears to be useful in tracking XP projects. However, students may not find it helpful due to the size and scope of their assignments. Additionally, the data recorded by Bryce can be used for research on XP. Currently, four active research projects are using data collected using the tool.

6.1 Future Research

6.1.1 Further tool Development

The deployment of Bryce in IBM and Genesys marked the first industrial releases of Bryce. As with any XP project, the development of Bryce needs to evolve with user feedback. Therefore development of Bryce is still an ongoing process. While substantial amount of Bryce development has been done, there are areas that need improvement and further development.

The users of Bryce have provided valuable suggestions for improving the tool. Some of the areas for improvement suggested by the students are:

- Better organization of the menu items and pages for easier navigation
- Reduce the number of mouse clicks for performing some operations like updating a user story, user task etc.
- Higher color contrast between text and the background for web pages

Other beneficial features for future releases of Bryce are listed below:

- Creation of a logging utility for logging the activities by the team members

- Automatic project data collection similar to the Hackystat tool [44] developed at the University of Hawaii. In the Hackystat tool, software "sensors" are attached to the developer's software tools and automatically send process data to a centralized web server. The web server integrates the streams of XML-based software engineering process data coming from the various sensors, produces just-in-time analyses, and emails the developer with URLs to these analyses.

7 List of References

1. Zahran, S. *Software Process Improvement.* Essex, England: Addison Wesley Longman Limited, 1998.
2. Royce, W.W. *Managing the Development of Large Software Systems: Concepts and Techniques*, Proceedings of *WESCON*, Vol. 14, pp. 1-9, August 1970.
3. Pressman, R.S. *Software Engineering: A Practitioner's approach.* 5th Edition. New York: The McGraw-Hill Companies Inc., 2001.
4. MacCormack, A. Product-Development Practices that Work: How Internet Companies Build Software in *MIT Sloan Management Review.* Vol. 42 Iss. 2, pp. 86-94, Winter 2001.
5. Connell, J.L. and L. Shafer. *Structured Rapid Prototyping: An Evolutionary Approach to Software Development.* Englewood Cliffs, New Jersey: Yourdon Press, 1989.
6. Boehm, B. A Spiral Model of Software Development and Enhancement in *IEEE Computer.* Vol. 11 Iss. 3, pp. 61-72, May 1988.
7. Nyberg, E.H. *Prototyping in E-Commerce Projects.* Carnegie Mellon University. 2002. http://hakata.mt.cs.cmu.edu/11-791/Handouts/Prototyping_rev02.pdf.
8. McConnell, S. *Rapid Development.* Redmond, Washington: Microsoft Press, 1996.

9. *The CHAOS Report.* The Standish Group International, Inc. 1994. http://www.standishgroup.com/sample_research/chaos_1994_1.php.

10. *CHAOS: A Recipe for Success.* The Standish Group International, Inc. 1999. http://www.standishgroup.com/sample_research/chaos1998.pdf.

11. Fowler, M. *The New Methodology.* June, 2002. http://www.martinfowler.com/articles/newMethodology.html.

12. Beck, K. *Extreme Programming Explained: Embrace Change.* Mass.: Addison-Wesley, 1999.

13. *DSDM consortium.* 2002. http://www.dsdm.org/.

14. Schwaber, K., M. Beedle, K. Schwaber, and R.C. Martin. *Agile Software Development with SCRUM.* N.J.: Prentice Hall, October 15, 2001.

15. Cockburn, A. *Agile Software Development.* Boston: Addison-Wesley, 2001.

16. Highsmith, J.A. *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems.* New York: Dorset House, January 2000.

17. Palmer, S.R. and J.M. Felsing. *A Practical Guide to Feature-Driven Development.* N.J.: Prentice Hall PTR, 2002.

18. Marchesi, M. Which AM should I Use? in *Extreme Programming Perspectives.* Boston, MA: Addison-Wesley. pp. 17-21, 2002.

19. Wells, D. *What is Extreme Programming?.* 2002. http://www.extremeprogramming.org/what.html.

20. Wells, D. *Do We Need Yet Another Methodology?.* 2002. http://www.extremeprogramming.org/another.html.

21. Williams, L.A. *The Collaborative Software Process.* Salt Lake City, UT: Phd Dissertation in University of Utah, 2000.

22. Williams, L.A. *Pair Programming Illuminated.* Boston, MA: Addison Wesley, 2002.

23. Ramachandran, V. and A. Shukla. Circle of Life, Spiral of Death: Are XP Teams Following the Essential Practices? in *Extreme Programming and Agile Methods - XP/Agile Universe 2002.* Berlin: Springer-Verlag. pp. 166-173, 2002.

24. Florac, W.A., R.E. Park, and A.D. Carleton. *Practical Software Measurement: Measuring for Process Management and Improvement.* SEI, Carnegie Mellon University. April, 1997. http://www.sei.cmu.edu/pub/documents/97.reports/pdf/97hb003.pdf.

25. Wells, D. *Project Velocity.* 2002. http://www.extremeprogramming.org/rules/velocity.html.

26. Jeffries, R., A. Anderson, and C. Hendrickson. *Extreme Programming Installed.* NJ: Addison Wesley, 2001.

27. Leuf, B. and W. Cunningham. *The Wiki Way.* N.J.: Addison-Wesley, 2001.

28. Maurer, F. Supporting Distributed Extreme Programming in *Extreme Programming and Agile Methods - XP/Agile Universe 2002.* Berlin: Springer-Verlag. 2002.

29. Chaffey, D. *Groupware, Workflow and Intranets: Reengineering the Enterprise with Collaborative Software.* MA: Butterwoth-Heinemann, 1998.

30. Cunningham, W. *Wiki Official Website.* 2002. www.wiki.org.

31. *Workflow.* 2002. http://webopedia.com/TERM/w/workflow.html.

32. *On-line demos.* 2002. http://sern.ucalgary.ca/~milos/milos/demos/demo.htm.

33. Lee, C.C. *JavaNCSS - A Source Measurement Suite for Java.* http://www.kclee.com/clemens/java/javancss/ 2002.

34. Succi, G. A Lightweight Evaluation of a Lightweight Process in *Extreme Programming Perspectives.* Boston, MA: Addison-Wesley. 2002.

35. *Java Server Pages.* 2002. http://java.sun.com/products/jsp/.

36. Clark, S. *WebDeveloper.com's Guide to Cascading Style Sheets.* 2002. http://www.webdeveloper.com/html/html_css_1.html.

37. *Caucho Technology.* 2002. http://www.caucho.com/.

38. Boehm, B.W. *Software Engineering Economics.* N.J.: Prentice-Hall, Inc., 1981.

39. Williams, L.A., G. Succi, M. Stefanovic, and M. Marchesi. A Metric Suite for Evaluating the Effectivness of an Agile Methodology in *Extreme Programming Perspectives.* Boston, MA: Addison-Wesley. pp. 335-347, 2002.

40. *What is JavaTM Technology?.* 2002. http://java.sun.com/java2/whatis/.

41. *Hypertext Markup Language - 2.0.* 2002. http://www.ietf.org/rfc/rfc1866.txt.

42. *MySQL Reference Manual for version 4.0.5-beta.* 2002. http://www.mysql.com/documentation/mysql/full/.

43. *JDBC Driver Certification Program.* 2002. http://java.sun.com/products/jdbc/certification.html.

44. *Hackystat (Research Summary).* 2002. http://csdl.ics.hawaii.edu/Research/Hackystat/.

Appendix 1: Installation Manual

Requirements

This version of the tool requires Windows 2000 to have been installed prior to this installation.

The other components to be installed are:

- Microsoft SQL Server 2000 (Refer to section 0 - Installation and Configuration of Microsoft SQL Server 2000 for more details)
- JDBC Driver for MS SQL server (Refer to section 0 - Setting up of JDBC driver for more details)
- Resin version 1.2.8 (Refer to section 0 - Installation and Configuration of Resin for more details)
- Internet Services Manager (Refer to section 0 - Installation and Configuration of IIS for more details)

To install the tool on Apache and Tomcat refer to section 0.

Extracting the compressed tool archive

Unzip Tool.zip using WinZip® in a directory where you wish to install the tool, for e.g. c:\wwwroot \. This directory will be the IIS web server root directory.

The following directories will be extracted:

- /XPTOOL - Contains the JSP files
- /WEB-INF - Contains the java classes
- /Dbconnect - Contains the JDBC vendor specific driver

- /Delete-Later - Miscellaneous files which should be deleted later

Installation and Configuration of Microsoft SQL Server 2000

1. Create a restricted user, “DBUser” with password “psp” in Windows 2000
2. SQL Server Setup - Follow the default setup except for the following
   - Choose "Local system for Login"
   - Choose Mixed Mode.
   - Password for account "sa" can be anything say "xyz"
3. Start SQL Server. Click on the Service Manager Icon.
4. Restore Databases
   - Create two Databases "User1" and "Data2" by right clicking on "Databases.
   - Overwrite them with databases User1 and Data2.
5. Restoring "User1" database
   - Click Enterprise Manager
   - Click Tools-> Restore Database
   - Select "From device" radio button
   - Click "Select Devices" -> "Add" ->"Filename" -> choose file User1
   - Click ok
6. Similarly restore database "Data2"
7. Setting up login
   - Remove DBUser from User1 database. Click on database User1->Users. Delete the user "DBUser" right clicking on it.

- Now click on Security->Logins. Right click to create new login.

In the “General” tab choose the following,

- Name: DBUser
- Check SQL Server Authentication. Enter password "psp". Windows authentication is unchecked.
- Database: Master
- Language: default

In the “Server Roles” tab,

- Click on System Admin and Database creators

In the “Database Access” tab,

- Choose Data2 and User1 .For both of these choose "public" and "db_owber"
- Re-enter password "psp" when prompted.

Setting up of JDBC driver

- Click on Start->Settings->Control Panel->System->Internet Services Manager->Advanced->Environment Variables
- Set CLASSPATH to "*.;c:\wwwroot\DBconnect\CaveConnectC.jar*" or wherever the file CaveConnectC.jar is located.

Installation and Configuration of Resin

- Unzip resin-1.2.8.zip using WinZip in a directory where you wish to store the tool, for e.g. c:\resin-1.2.8

- Replace Resin.conf in c:\resin-1.2.8\conf\Resin.conf with c:\wwwroot\Delete-Later\Resin.conf
- Go to c:\resin-1.2.8\bin and execute "srun -install" to install srun as an NT service.
- Click on Start -> Settings -> Control Panel -> Administrative Tools -> Services. Start Resin by right clicking on the Resin Service and clicking on start.
- Now the Resin service has been started.

Installation and Configuration of IIS

- Install the IIS using the Windows 2000 CD.
- Click on Start->Settings->Control Panel->Administrative Tools->Internet Services Manager. IIS window should open up. If the Internet Services Manager icon is not present, then IIS is not setup or has not been setup properly.
- In the tree window on the left hand side, click on your computer icon and expand the tree. “Default Website” icon should be visible.
- Setting the server home directory for the default Website:
- Right click on Default Website and click on properties.
- Now slick on the home directory heading.
- Set the home directory (for e.g. c:\wwwroot).

- Create a virtual directory for the tool
  - Right click on “Default Website” and click on properties.

- Now Click on New->Virtual directory. The virtual directory wizard opens up. Click next.
- For Virtual Directory Alias enter "XPTool". Click next.
- For the Website Content Directory enter the tool directory (for e.g. c:\wwwroot\xptool). Click next.
- For Access Permissions, check "Read" & "Run Scripts". Click next.
- Click finish.
- Right click on “Default Website” and click on start.
- Now the website is online.

- Now delete the directory /Delete-Later which was extracted during the installation.
- Access the tool using http://hostname/xptool/default.jsp in the Web browser.

Installation on Apache and Tomcat

- Set up MS SQL server and the JDBC driver in the same way as mentioned above
- Install Apache Version 2.x and Tomcat 4.x
- Create a context for the tool in tomcat. Insert the following in $TOMCAT_HOME/conf/server.xml

  <Context path="/XPTOOL" docBase="XPTOOL" debug="0" reloadable="true" crossContext="true"/>

- Move WEB-INF directory inside XPTOOL directory

- Place the XPTOOL directory in $TOMCAT_HOME/webapps/
- Access the Tool using the URL http://hostname:8080/XPTOOL/default.jsp

Trouble Shooting

The site is not accessible

- Check the URL of the tool
- Check if the IIS Web server is running by clicking on Start->Settings->Control Panel->Administrative Tools->Services. IIS status should have been started.
- Check if Resin is running by clicking on Start->Settings->Control Panel->Administrative Tools->Services. Resin status should have been started.
- In the web browser go to http://localhost/xptool/test.jsp
- It should display 2+2=4. This shows that the JSP engine is running.
- Also, go to http://localhost/caucho-status for the status and configuration of Resin.

Could not connect to the database

- Check the environment variable path of the JDBC driver.
- Check if the database has been installed properly.
- Check for database permissions. The username must be “DBUser” and the password must be “psp” for the databases Data2 and User1. Also, “DBUser” should have “public” and “db_Owner” permissions. If any modifications are made, restart the MS SQL server.

APPENDIX 2: User Manual

Bryce is a tool specially developed for managing XP projects.

Overview

The tool can be accessed at http://bryce.csc.ncsu.edu/XPTool/default.jsp. To use the tool, a user has to first register to the system. The user can register to the system using the "New user signup" link in the login page. Once the new user has registered to the system, he can login to the tool. There are two types of users

- Normal user – A normal user is one who can use the tool to record/view metrics data with restricted privileges. He can
    - Add User Stories and Tasks pertaining to the Project
    - View/Record certain Project information
    - Perform limited Administration tasks
- Manager: A manager has certain special privileges to the system in addition to the normal user privileges. The manager can also
    - Add/Delete Projects
    - Add/Delete/Edit Release information
    - Add/Delete/Edit Iterations of a Release
    - Add/Delete/Edit User Stories and Tasks pertaining to the Project

For information regarding installation of the tool, please refer to Install.doc.

New User Registration

In order to use the tool, a user must be registered to the system. The new user signup is used to register a new user to the system. The user is required to enter the following fields in the specified format:

- User ID/Email-Address
- Password
- Confirm Password
- First name
- Last name
- Manager (choose your manager from the drop down list)

Once a user has successfully registered to the system, he can login with the Email Id and the password he had specified during the registration.

Description of the Menu items

Firstly, the user has to select a project from the dropdown menu. Then the following submenu items will then be available to the user.

- **My Stuff**
  - **My Stories:** This sub menu item is used to view user stories assigned to the current user.
  - **My Tasks:** This sub menu item is used to view user tasks assigned to the current user.
  - **Change My Story Status:** This sub menu item is used to change the status of the story. The story states can be “not done”, “done”, “pre-

release change", "post release change", "refactoring" and "bug fix change".

- **Change My Task Status:** This sub menu item is used to change the status of the task to completed.
- **Record Task Time:** This sub menu item is used to record time to user tasks assigned to the current user. The user specifies the amount time spent on the particular task and the collaborators involved in the task.
- **Edit Task Time:** This sub menu item is used to edit the recorded time spent on user tasks.

- **Feature Management**
  - **Add Stories:** This sub menu item is used to add new user stories to the project. The user enters a unique new story identifier, which is an alphanumeric string (without spaces, special characters), in order to identify this story. This story can be added to a release or iteration during a planning game.
  - **Add Task:** This sub menu item is used to add new user tasks, which belongs to a particular story in the project. The user enters a unique new task identifier, which is an alphanumeric string (without spaces, special characters), in order to identify this task.
  - **Add Acceptance Tests:** This sub menu item is used to add new acceptance tests information. Each acceptance test is associated with a user story.

- **Change Story Status:** This sub menu item is used to change the status of the story. The story states can be “not done”, “done”, “pre-release change”, ”post release change”, “refactoring” and “bug fix change”.
- **Track Story Change:** This sub menu item is used to view all the changes that have been made to the story.
- **View/Edit Stories:** The current user uses this menu item to view/edit user stories and tasks. He can view the tasks corresponding to a particular story by clicking on the story. The “Edit” buttons beside the stories and tasks are used to edit the corresponding stories and tasks. Once the “Edit” button is clicked the individual fields of the story or task can be edited. Also, stories and tasks can be deleted clicking the “Delete” button beside them.
- **View/Edit Tasks:** The current user uses this menu item to view/edit user tasks. Once the “Edit” button is clicked the individual fields of the task can be edited. Also, the tasks can be deleted clicking the “Delete” button beside them.
- **View/Edit Acceptance Tests:** The current user uses this menu item to view/edit Acceptance tests (AT). Once the “Edit” button is clicked the individual fields of the AT can be edited. Also, the AT’s can be deleted clicking the “Delete” button beside them.

- **Project Information**

This is used for recording the time for the different phases of the project. The submenu items are

- **Project Summary:** This submenu item gives information regarding the current project.

- **Administration**

This menu item is used for the administration of the project. The submenu items are

- **Add Team Member:** This submenu item is used to add a new team member to the project. The new team member must have registered to the system before he can be added as a team member to the project. This menu item displays the current team members of the project. To add a new team member, the username and the password must be entered. The user is required to enter the following fields in the specified format:
  - User ID (Email id used during registration)
  - Password
- **Delete User Registration:** This submenu item is used to delete a user from the system. This menu item removes the currently logged in user from the system. To delete a user, the password must be entered. The user is required to enter the following fields in the specified format:
  - Password

- **Add/Change Manager:** This submenu item is used to change the current project manager. The user can choose a new manager from a list of project managers registered to the system.

- **Change Password:** This submenu item is used to change the password of the current user.

- **Manager Menu (this menu is only visible for a manager)**

This menu item is only visible to a Manager. The manager to perform certain privileged tasks uses this menu item. The submenu items are:

- **Create Projects:** The manager, to add a new project to the system, uses this menu item. The manager specifies the new project name, which is an alphanumeric string without spaces and special characters. The user is required to enter the following fields in the specified format:
  - Project Name (alphanumeric string without spaces, special characters)
- **Delete Project:** The manager, to delete a project from the system, uses this menu item. The manager will be asked to confirm the deletion. Once a project has been deleted the entire project related information is lost forever and cannot be recovered.
- **Create Release:** The manager, to add a new release to the project, uses this menu item. The manager specifies the new Release-Identifier, which is an alphanumeric string without spaces and special characters and a

headline. The user is required to enter the following fields in the specified format:

- New Release Identifier (alphanumeric string without spaces, special characters)
- Release Headline (A brief headline regarding the release)

- **Delete Release:** The manager, to delete release of the project, uses this menu item. The manager selects Release-Identifier, which is to be deleted. The manager will be asked to confirm the deletion.
- **Add Iterations:** The manager, to add a new iteration to the release, uses this menu item. The manager specifies the new Iteration-Identifier, which is an alphanumeric string without spaces and special characters and a headline. The user is required to enter the following fields in the specified format:
  - New Iteration Identifier (alphanumeric string without spaces, special characters)
  - Choose the Release ID
  - Iteration Comments (A brief headline regarding the iteration)
- **Delete Iterations:** The manager, to delete iteration of a release, uses this menu item. The manager selects Iteration-Identifier, which is to be deleted. The manager will be asked to confirm the deletion.

- **Test/Debug**

  This menu item is used for debugging purposes.

- **Logout**

  This menu item is used to log out of the system.

How do I use this Tool to track my Project?

1. Create a new project in the tool. You need manager privileges to create a new project. This is done using the “create project” menu item.
2. Create Releases and Iterations in the tool. Again, you need manager privileges to do so. This is done using the “create release”& ”create iteration” menu items.
3. Determine the set of features the product/project needs to have. Each of these are called “User Stories” in XP. Enter the User Stories in the tool. This is done using the “Add User Story” menu item.
4. Create acceptance test cases for each of these user stories. Enter the acceptance test case information in the tool. This is done using the “Add Acceptance Test” menu item.
5. Play the “Planning game”: For each of these stories, estimate the time and assign priority. Record this information using the “View /Edit User Stories” menu item.
6. Each User story is divided into tasks and each task & story is assigned a responsible owner. Record all this information using the “Add Task”, “View /Edit User Stories” and “View /Edit Tasks” menu items.
7. Design a simple architecture for the product (XP philosophy).

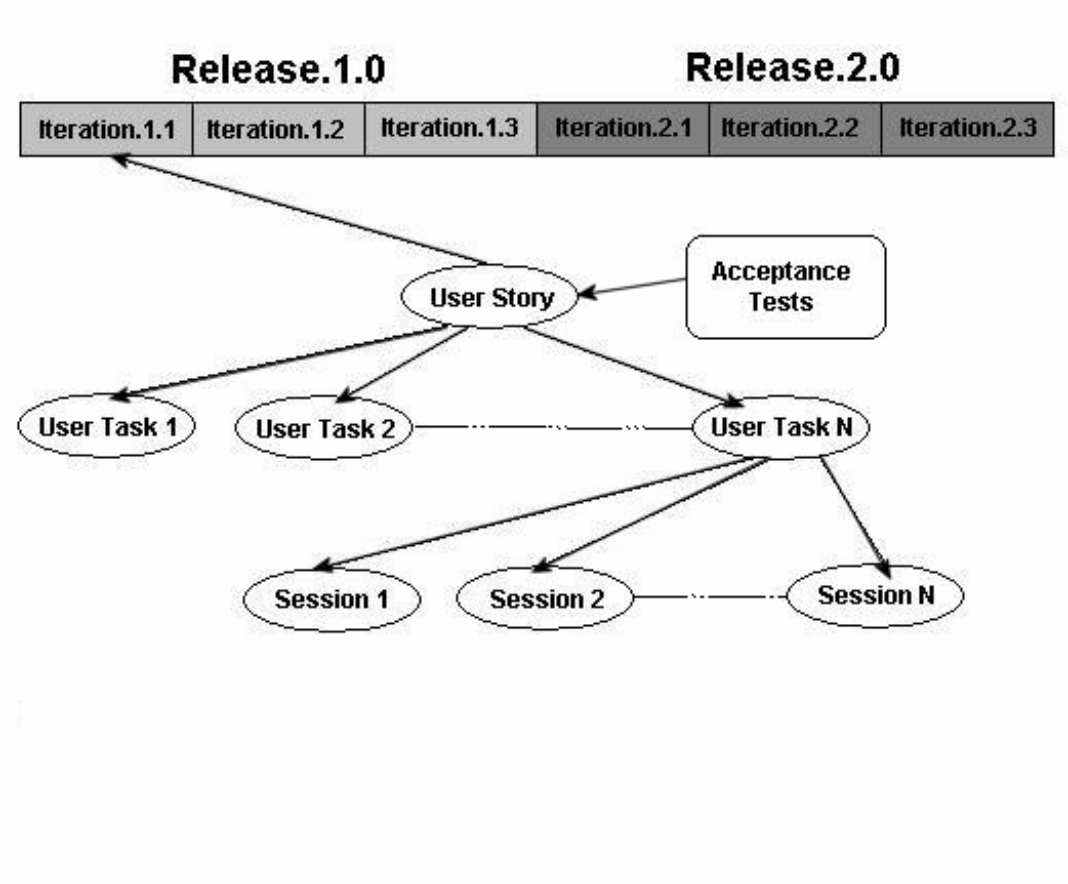
8. Implement each of these User Stories/Tasks based on the priority. The design and implementation for each of these tasks is done in pairs (Pair programming). Keeping track of time spent on each task can be done using “Record Task Time” menu item.
9. Perform unit testing after the completion of each task and update the status of the task as “completed” in the tool. This is done using the “Change My Task Status” menu item.
10. Once all the tasks of the user story have been implemented, run the acceptance test cases for the user story. If the acceptance test passes, then change the acceptance test status to “Pass” and story status to “Done” in the tool. This is done using the “View /Edit Acceptance Tests” and “Change My Story Status” menu items.
11. Complete all the user stories for a release and record all the information as mentioned above.

APPENDIX 3: Tutorial

Setting up the tool for your project

7.1 Introduction

Bryce is a tool for managing XP projects. The Bryce tool facilitates project management by means of simple data entry and provides elaborate reporting on project status and progress. The tool will provide automated support for project management, process data gathering, and process metric analysis. The following figure illustrates the simplistic view of an XP project



In XP, the user stories represent the features to be implemented in the project. As shown in the figure, each of these user stories is divided into one or more user tasks. Pair programmers implement each of these tasks in a session. The acceptance tests are written to verify the user stories.

Additionally, a project is shipped to a customer in short release cycles. Each release contains a set of product features (or user stories). Also, each release is broken down into one or more iterations, which typically lasts two to four weeks. One or more user stories are implemented for each iteration.

Getting Started

The Tutorial is based on the development of this sample application/product

“Library software that keeps track of its members and its books"

We will call this project LibSoft. The developers have identified four features (called user stories in XP) that have to be present in the product

- AddNewMembers - Add new members to the library
- AddNewBooks – Add new books to the library
- BookCheckout – Record that a book is checked out by a member
- ShowBooksofMember – List the books a member has been borrowed by a member

Each of the following stories is split up into one or more tasks. For example, the AddNewBooks Story is split into the CreateBooksDatabase and CreateWebInterface tasks. The BookCheckout Story has one task CreateWebCheckOut, etc.

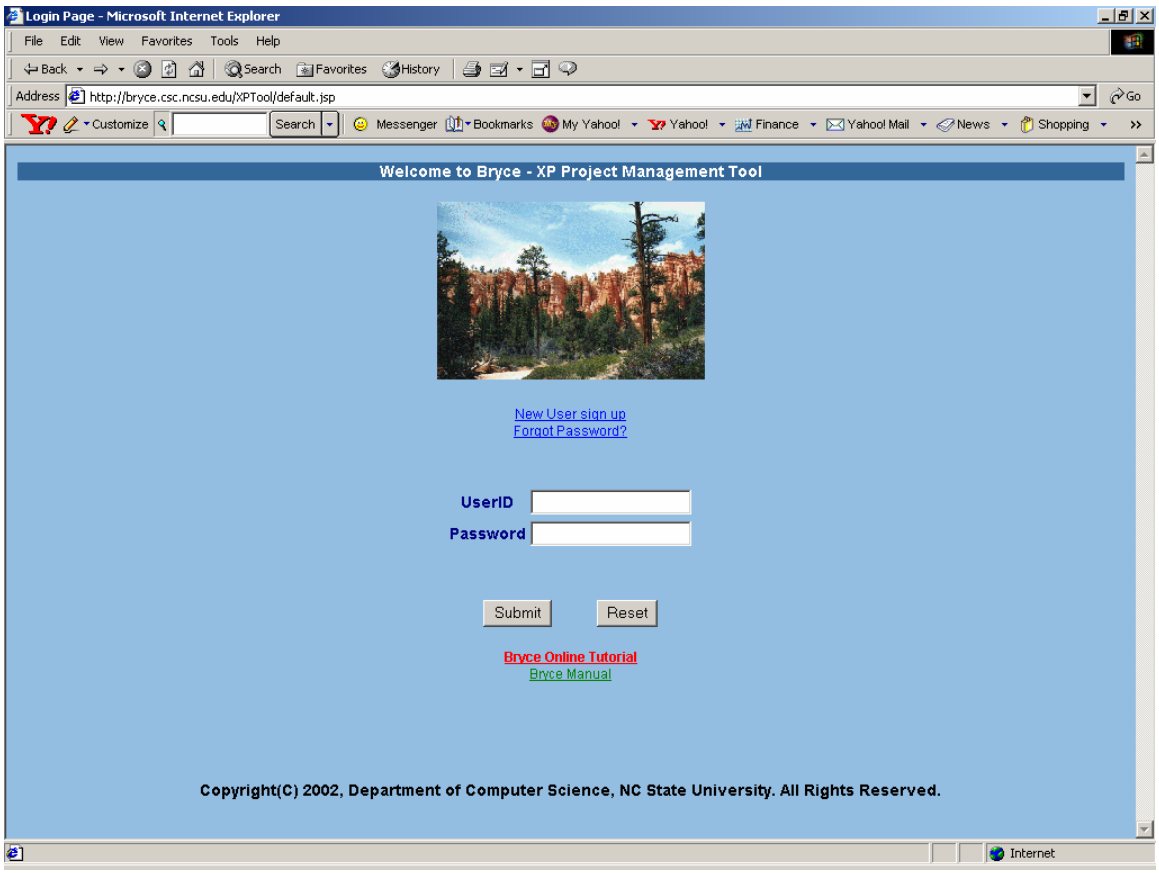
Additionally, the product is scheduled to be released in Release.1.0. The Release will contain two iterations called Iteration.1.0 and Iteration.2.0. The stories AddNewMembers and AddNewBooks will go into Iteration.1.0 & Stories BookCheckout and ShowBooksofMember will go into Iteration.2.0

Requirements

The tool requires Internet Explorer 5+ or Netscape 6.2+ browser.

Accessing the XP tool

The tool can be accessed at http://bryce.csc.ncsu.edu/XPTool/default.jsp. To use the tool, a user has to first register to the system. The user can register to the system using the "New user signup" link in the login page. Once the new user has registered to the system, he can login to the tool.



New User Registration

In order to use the tool, a user must be registered to the system. The user can register to the system using the "New user signup" link in the login page.

To register a user named David Wilson, enter the email address (which is also the User Id), password and choose a manager.

Upon successful registration, a mail will be sent to the manager stating that a new user “David Wilson” has registered to the tool with him/her as the manager. The manager has to add the new user to the project team in order for the new user to access project information.

In case, the user has forgotten the password, the user can click on the Forgot Password” link in the login page and the password will be mailed to the user’s email address.

The following screen shows how to register a user named David Wilson. In this tutorial, the following information was entered in this page and the submit button was clicked to register the user.

UserId: david@yahoo.com

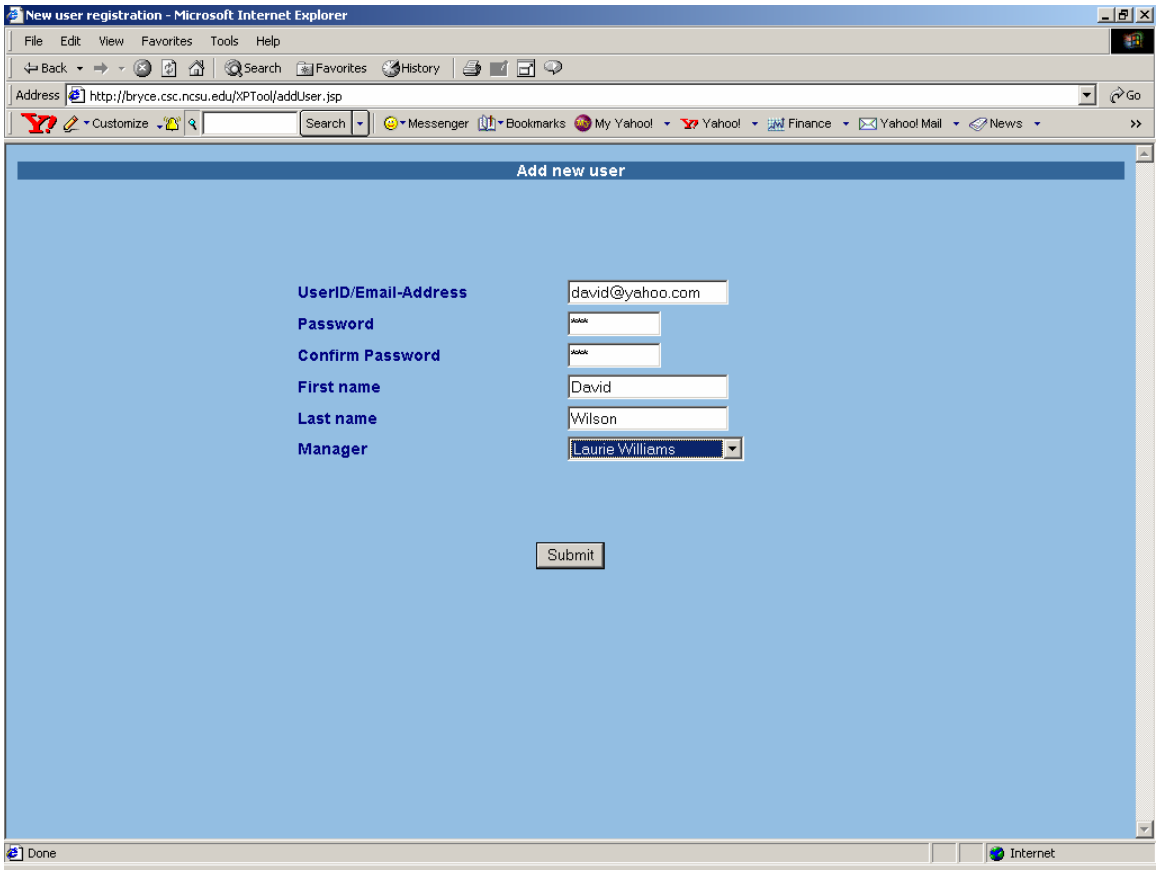
First Name: David

Last Name: Wilson

Password: ***

Confirm Password: ***

Manager: Laurie Williams



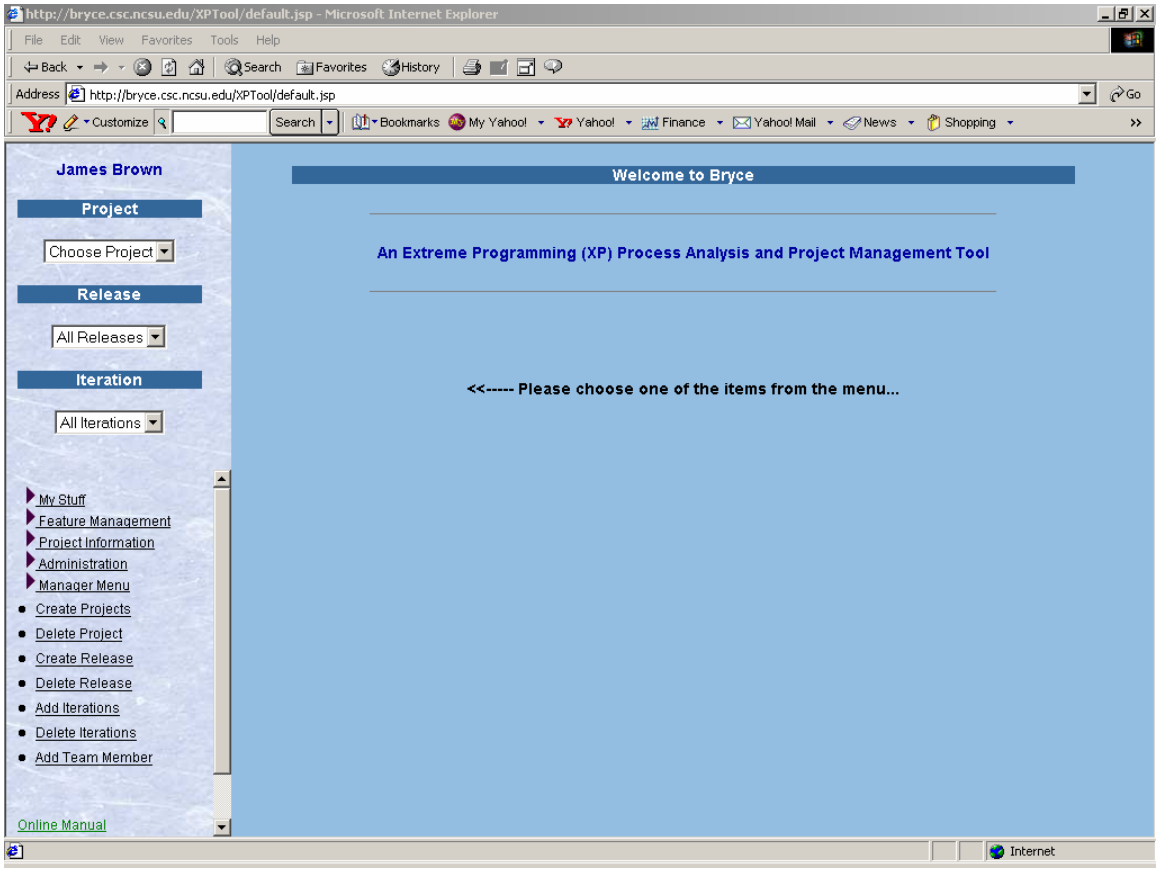
Once a user has successfully registered to the system, he can login with the User Id and the password he had specified during the registration.

Once a user has registered, he must do one of the following

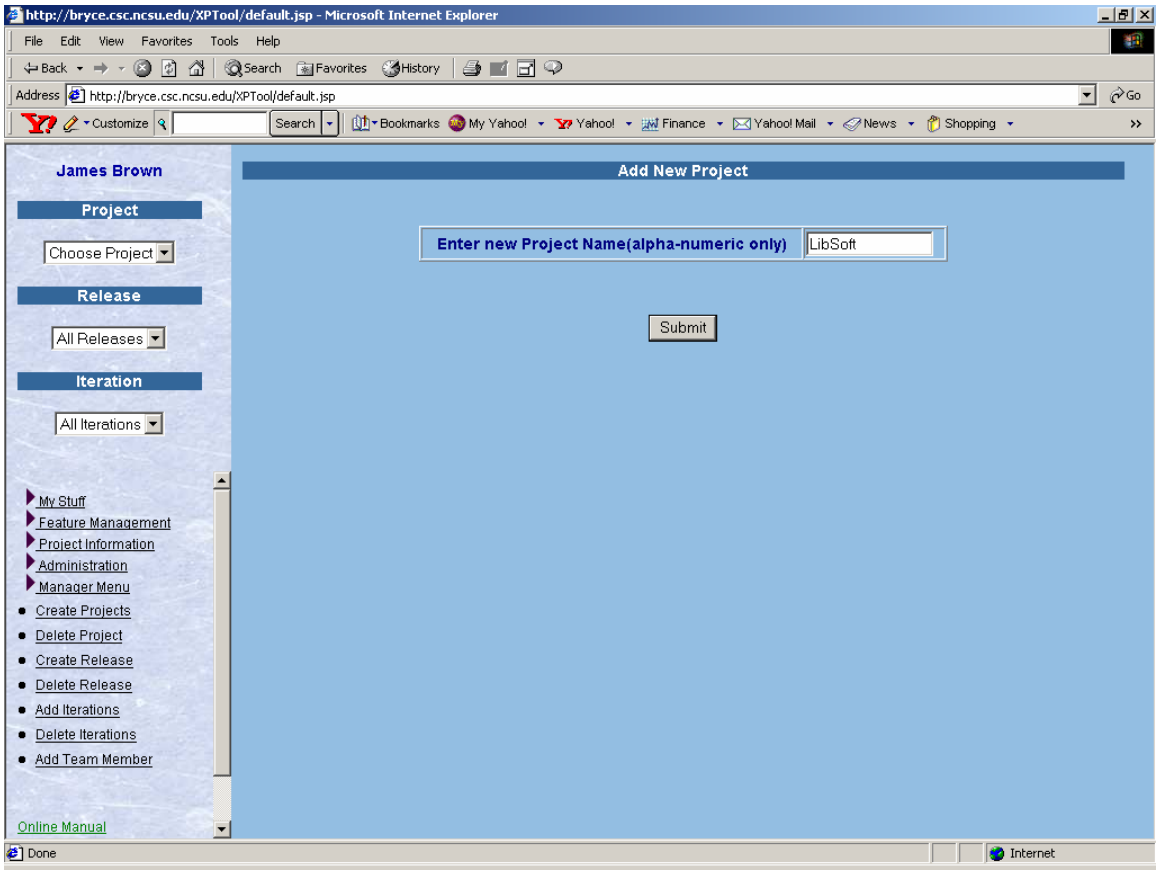
- **Add a Project** - To do so, the user must be a manager. Only the system administrator can change a user to a manager.
- **Get yourself added to an existing project** – To do so, the manager under whom the user signed up under must add the user to the project.

Adding New Project (Manager Only)

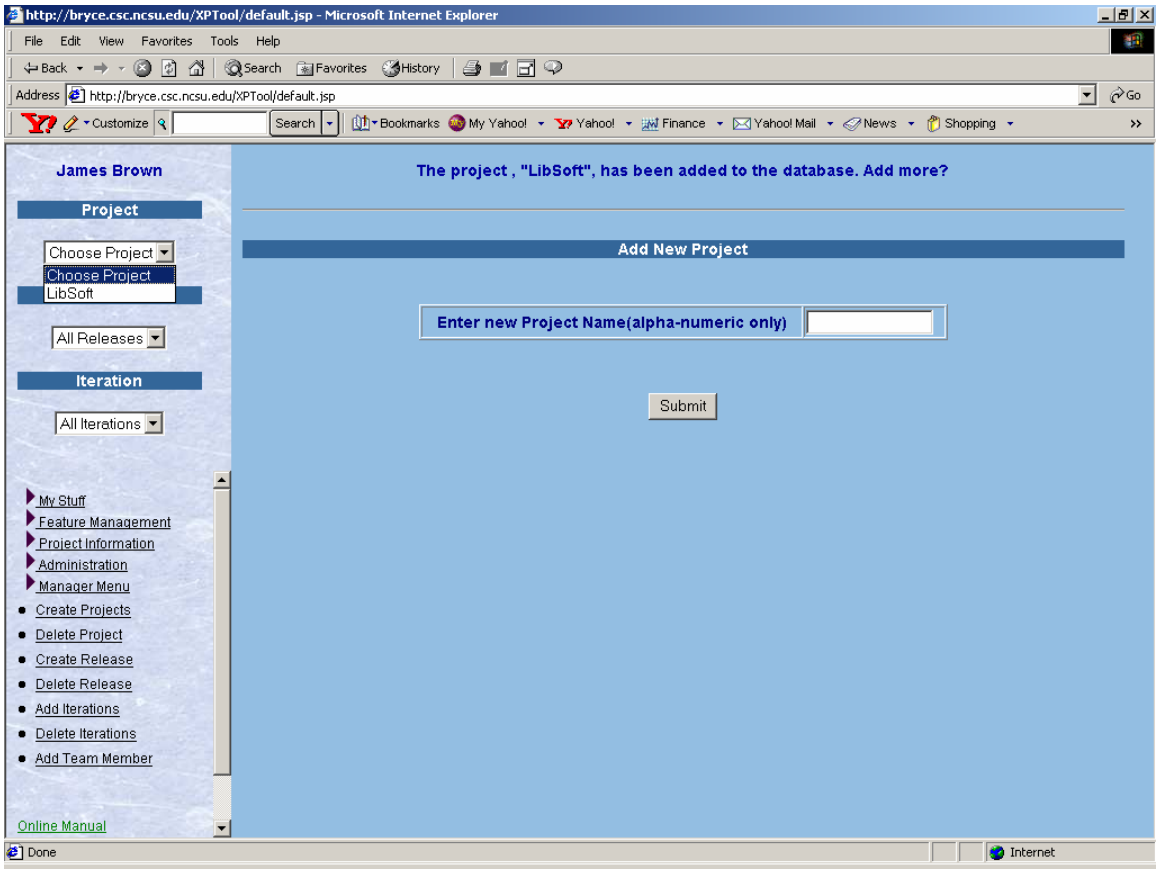
Once a user has been added as a manager, he/she can view the Manager Menu. Only the manager has access to the manager menu items.



Now click on the **"Manager Menu->Create Project"** and enter the project name LibSoft as shown below. Then click on the submit button.


http://bryce.csc.ncsu.edu/XPTool/default.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://bryce.csc.ncsu.edu/XPTool/default.jsp
Go
Customize Search Bookmarks My Yahoo! Yahoo! Finance Yahoo! Mail News Shopping
James Brown
Project
Choose Project
Release
All Releases
Iteration
All Iterations
My Stuff
Feature Management
Project Information
Administration
Manager Menu
Create Projects
Delete Project
Create Release
Delete Release
Add Iterations
Delete Iterations
Add Team Member
Online Manual
Add New Project
Enter new Project Name(alpha-numeric only)
LibSoft
Submit
Done
Internet

On clicking submit, the project LibSoft is created. Now select LibSoft from the drop down menu (left top corner).



Adding Release, Iteration (Manager Only)

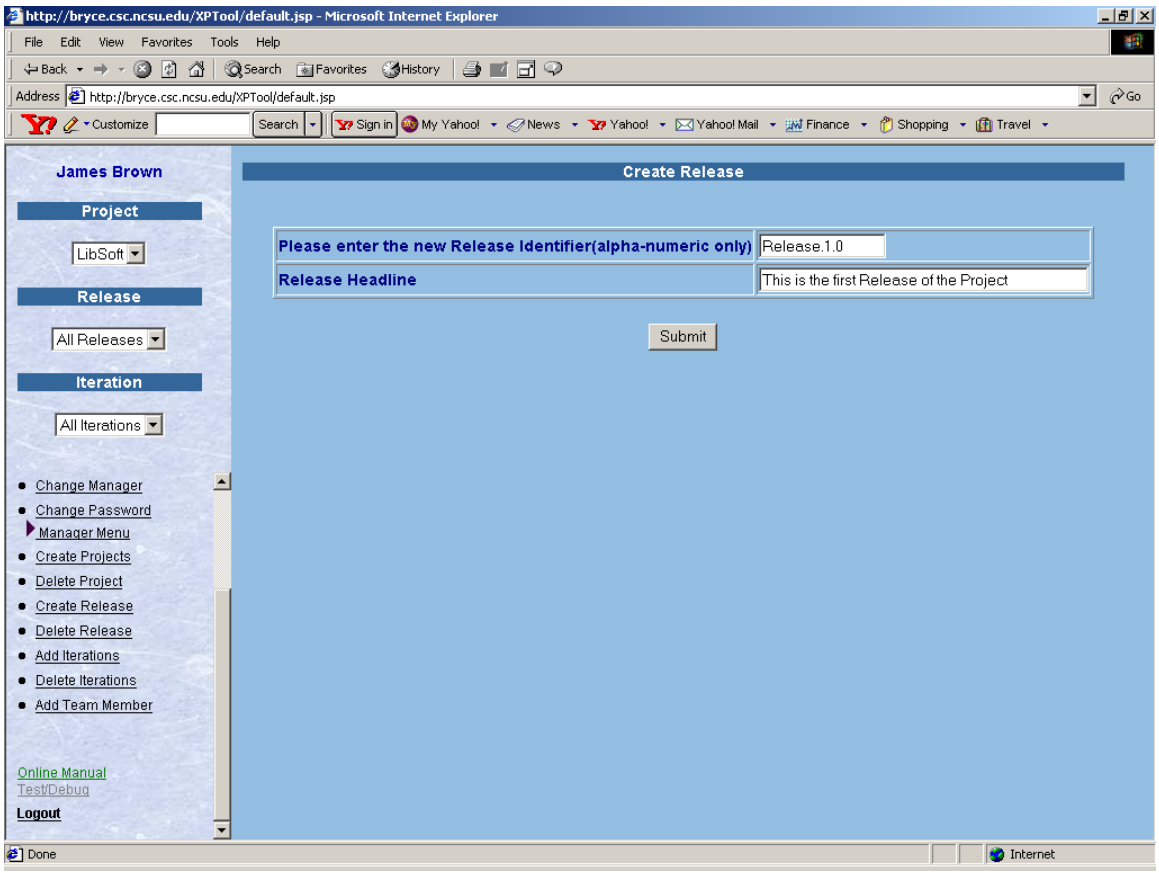
A manager can create release and iterations by clicking on the "**Manager Menu->Create Release**" and "Add Iterations" menu items respectively.

- Ensure that the project LibSoft is selected in the projects dropdown menu
- To add a Release, the manager specifies the new Release-Identifier, which is an alphanumeric string without spaces and special characters and a headline

The following screen shows how to add Release.1.0. In this tutorial, the following information was entered in this page and the submit button was clicked.

Release Id (alpha-numeric only): Release.1.0

Release Headline: This is the first Release of the product

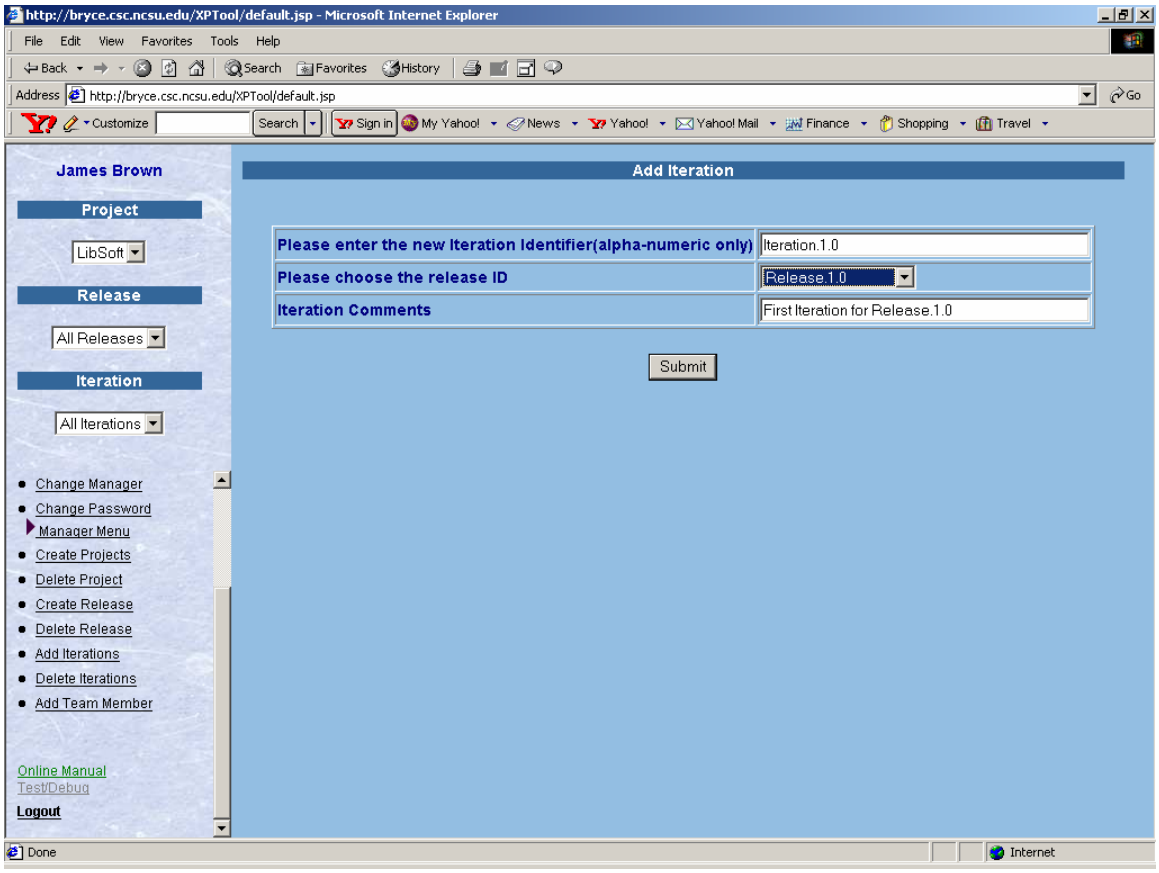


The following screen shows how to add Iteration.1.0 and Iteration.2.0. In this tutorial, the following information was entered in this page and the submit button was clicked.

Please enter the new Iteration Identifier (alpha-numeric only): Iteration.1.0

Please choose the release ID: Release.1.0

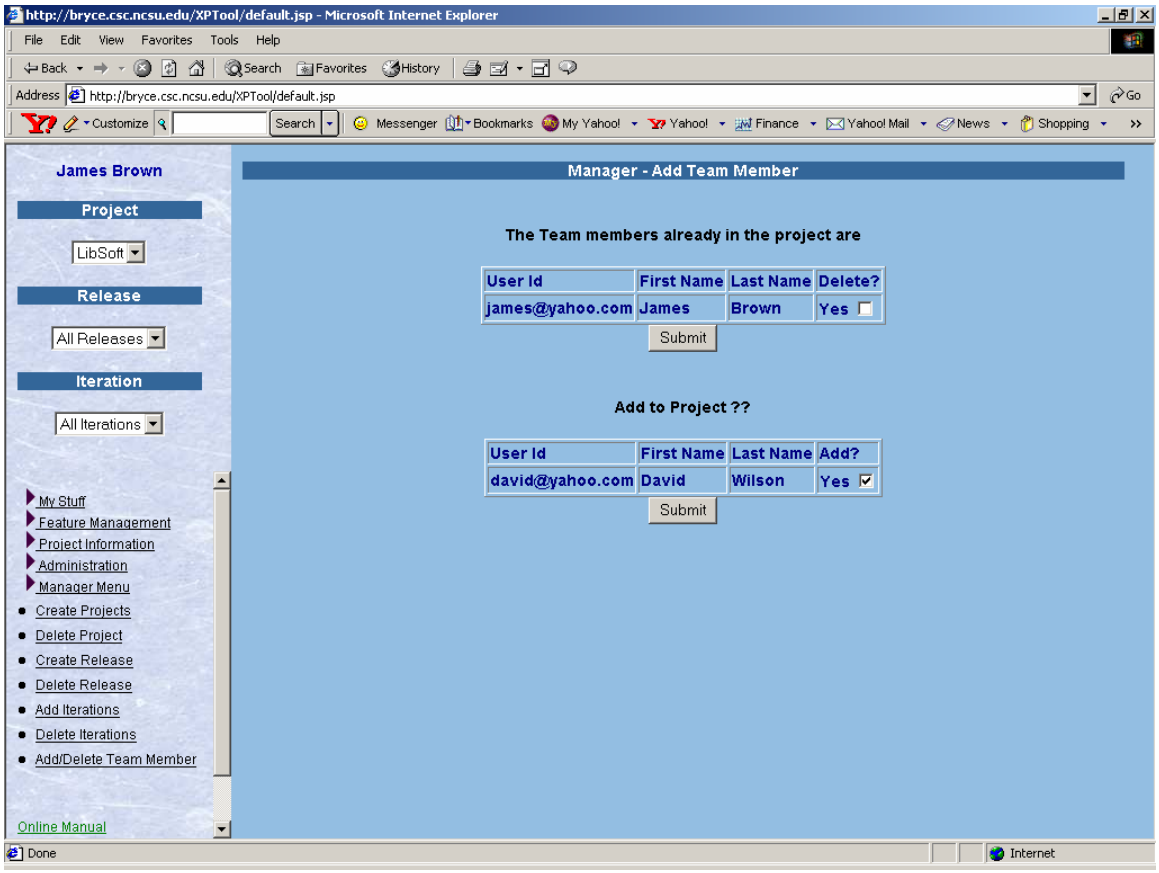
Iteration Comments: First Iteration for Release.1.0



Adding/Deleting New Team Member (Manager Only)

A manager can add new team members by clicking on the "**Manager Menu->Add/Delete Team Member**" menu item. To do so,

- Ensure that the project LibSoft is selected in the projects dropdown menu
- The current team members of the project are displayed.
- To add a new Team member David, the manager selects him in the list and clicks on the submit button.



Feature Management

Adding a new story

A user can add stories by clicking on the "**Feature Management->Add Story**" menu item. The Story AddNewMembers is added as shown below.

- Ensure that the project LibSoft is selected in the projects dropdown menu
- Enter new story-identifier, which is an alphanumeric string without spaces and special characters

  Story Identifier (alpha-numeric only): AddNewMembers
- Choose the release and iterations from the respective dropdown menus

Release ID: Release.1.0

Iteration: Iteration.1.0

- Also choose the developer risk and customer priority

  Customer Priority: 0(Lowest)

  Developer Risk: Moderate

- Enter the starting date for the user story in mm/dd/yyyy format

  Story Start Date (MM/DD/YYYY): 08/15/2002

- Enter the estimated time to complete the story in hrs

  Estimate (in hrs): 15.0

- Enter the story description

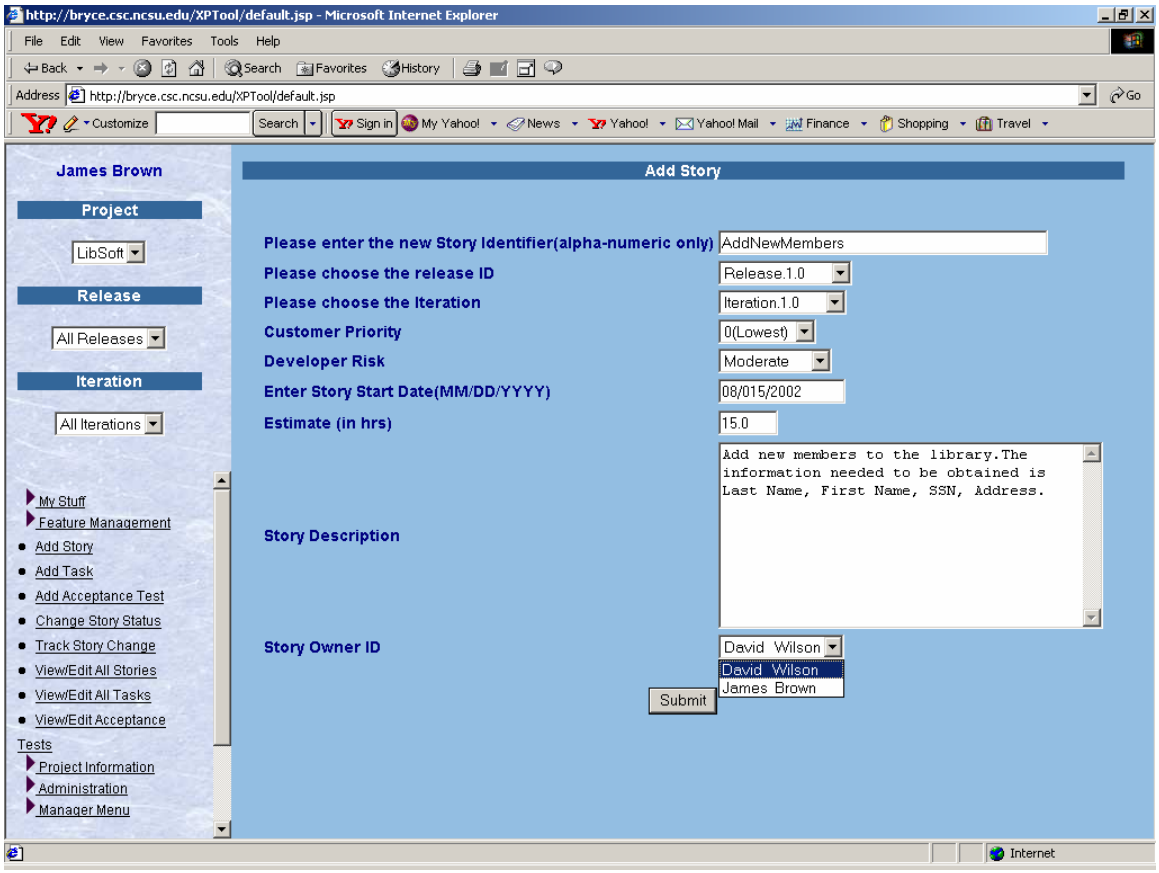
  Story Description: Add new members to the library. The information needed to be obtained is Last Name, First Name, SSN, Address.

- Finally assign the story to a team member

  Story Owner ID: David Wilson

If the user is unsure of what to enter or has made a mistake when entering the data, he/she can change them later using "**Feature Management->View/Edit All Stories**" menu item.

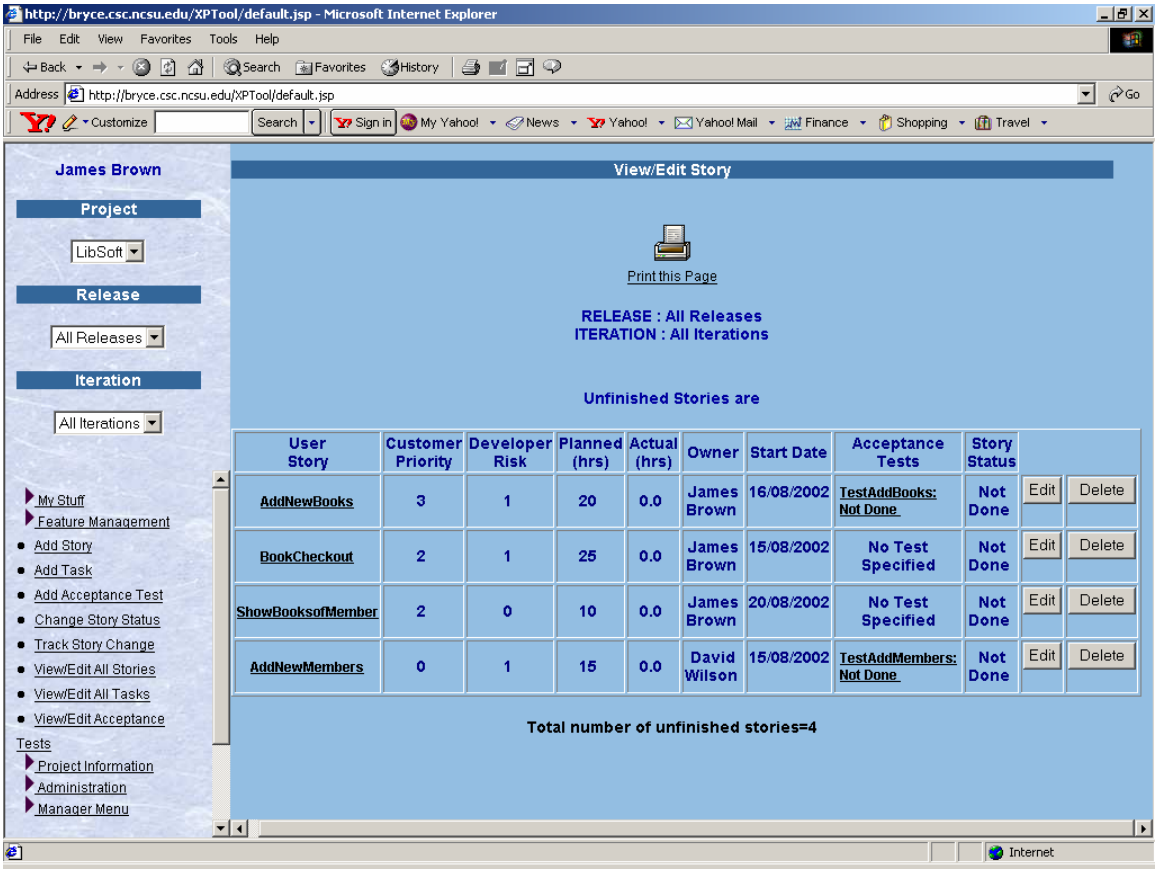
The following screen shows how to add AddNewMembers user story. In this tutorial, the following information was entered in this page and the submit button was clicked.


http://bryce.csc.ncsu.edu/XPTool/default.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://bryce.csc.ncsu.edu/XPTool/default.jsp
Go
Customize Search Sign in My Yahoo! News Yahoo! Yahoo! Mail Finance Shopping Travel
James Brown
Project
LibSoft
Release
All Releases
Iteration
All Iterations
My Stuff
Feature Management
Add Story
Add Task
Add Acceptance Test
Change Story Status
Track Story Change
View/Edit All Stories
View/Edit All Tasks
View/Edit Acceptance Tests
Project Information
Administration
Manager Menu
Add Story
Please enter the new Story Identifier(alpha-numeric only)
AddNewMembers
Please choose the release ID
Release.1.0
Please choose the Iteration
Iteration.1.0
Customer Priority
0(Lowest)
Developer Risk
Moderate
Enter Story Start Date(MM/DD/YYYY)
08/015/2002
Estimate (in hrs)
15.0
Story Description
Add new members to the library.The information needed to be obtained is Last Name, First Name, SSN, Address.
Story Owner ID
David Wilson
David Wilson
James Brown
Submit
Internet

Viewing and editing all stories

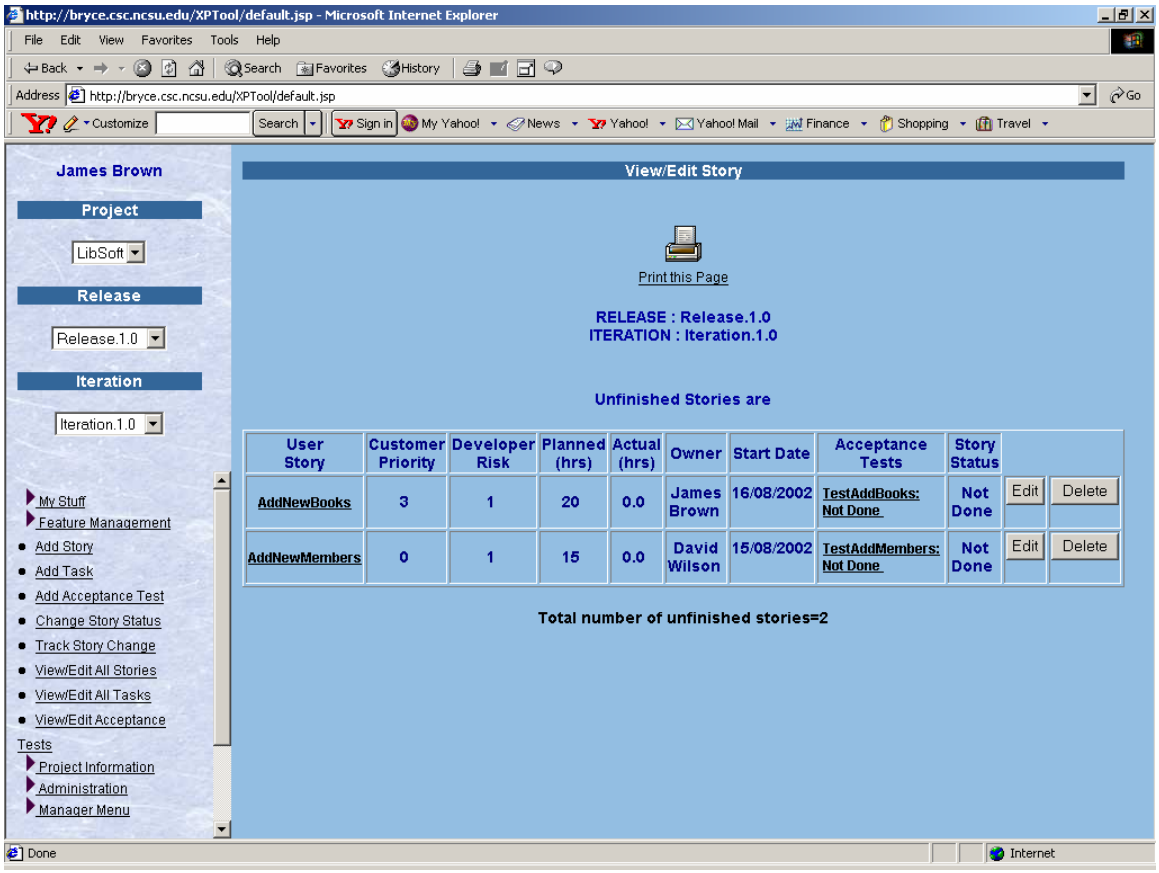
For viewing all stories in all releases and iterations,

- Select “All Releases” in the release drop down menu and “All Iterations” from the iteration drop down menu
- Now click on “**Feature Management->View/Edit All Stories**” menu item

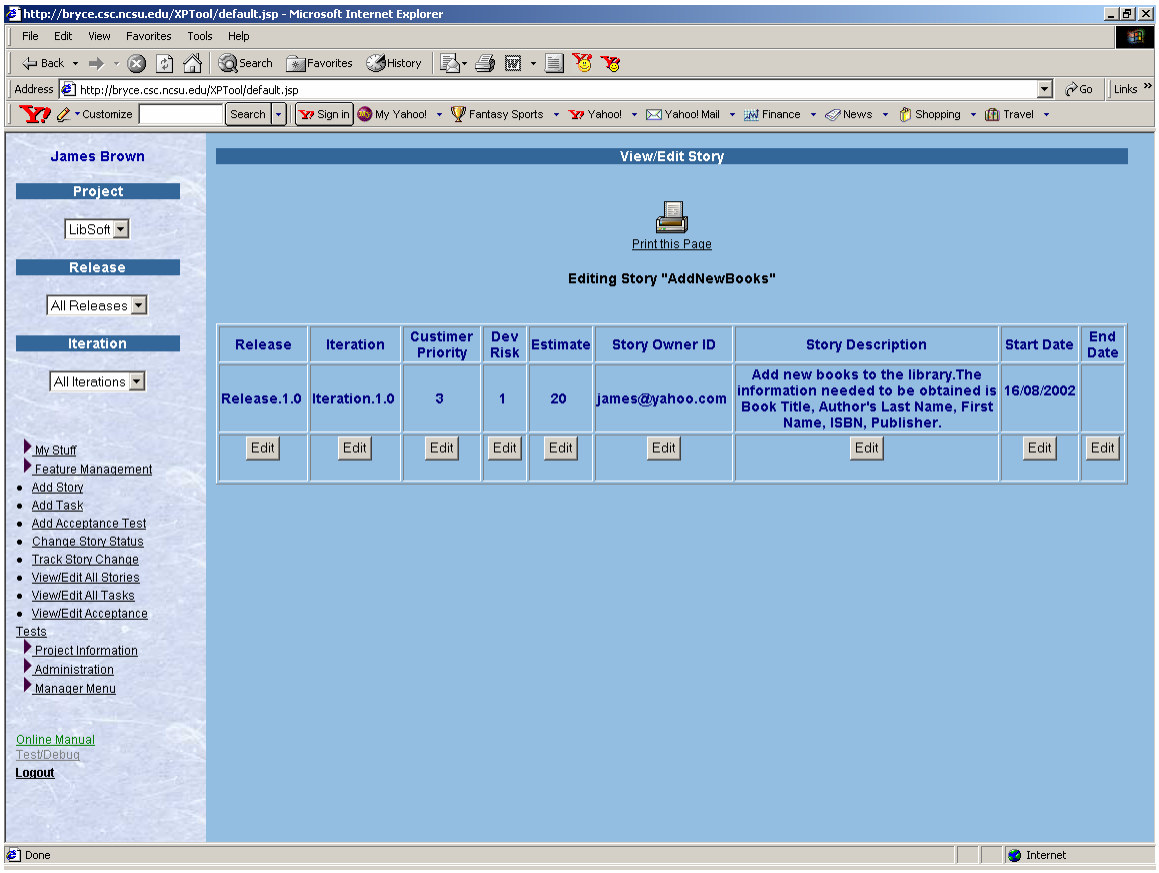

http://bryce.csc.ncsu.edu/XPTool/default.jsp - Microsoft Internet Explorer
James Brown
Project
LibSoft
Release
All Releases
Iteration
All Iterations
My Stuff
Feature Management
Add Story
Add Task
Add Acceptance Test
Change Story Status
Track Story Change
View/Edit All Stories
View/Edit All Tasks
View/Edit Acceptance Tests
Project Information
Administration
Manager Menu
View/Edit Story
Print this Page
RELEASE : All Releases
ITERATION : All Iterations
Unfinished Stories are
User Story | Customer Priority | Developer Risk | Planned (hrs) | Actual (hrs) | Owner | Start Date | Acceptance Tests | Story Status
AddNewBooks | 3 | 1 | 20 | 0.0 | James Brown | 16/08/2002 | TestAddBooks: Not Done | Not Done | Edit | Delete
BookCheckout | 2 | 1 | 25 | 0.0 | James Brown | 15/08/2002 | No Test Specified | Not Done | Edit | Delete
ShowBooksofMember | 2 | 0 | 10 | 0.0 | James Brown | 20/08/2002 | No Test Specified | Not Done | Edit | Delete
AddNewMembers | 0 | 1 | 15 | 0.0 | David Wilson | 15/08/2002 | TestAddMembers: Not Done | Not Done | Edit | Delete
Total number of unfinished stories=4

For viewing all stories in a specific release/iteration (for e.g. Release.1.0, Iteration.1.0)

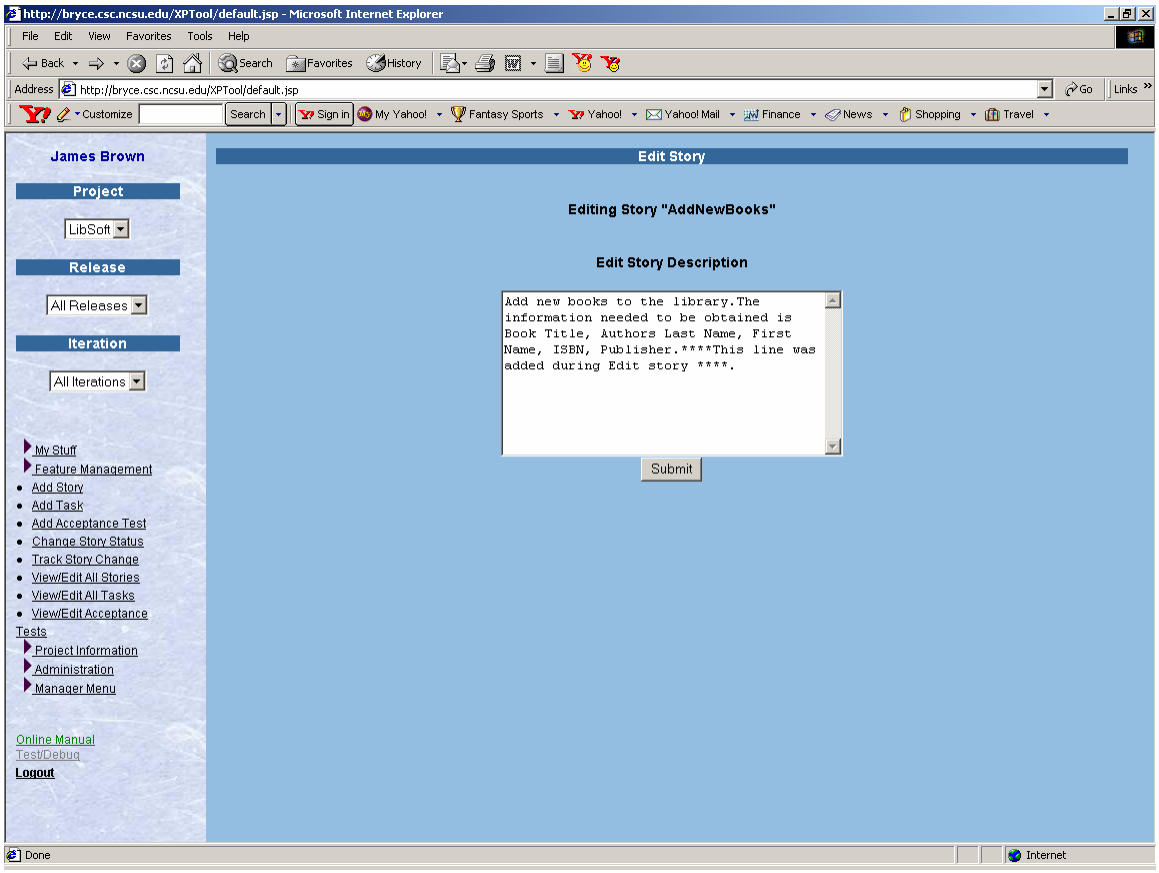
- Select Release.1.0 in the release drop down menu and Iteration.1.0 from the iteration drop down menu
- Now click on “**Feature Management->View/Edit All Stories**” menu item


http://bryce.csc.ncsu.edu/XPTool/default.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://bryce.csc.ncsu.edu/XPTool/default.jsp
Customize Search Sign in My Yahoo! News Yahoo! Yahoo! Mail Finance Shopping Travel
James Brown
Project
LibSoft
Release
Release.1.0
Iteration
Iteration.1.0
My Stuff
Feature Management
Add Story
Add Task
Add Acceptance Test
Change Story Status
Track Story Change
View/Edit All Stories
View/Edit All Tasks
View/Edit Acceptance Tests
Project Information
Administration
Manager Menu
View/Edit Story
Print this Page
RELEASE : Release.1.0
ITERATION : Iteration.1.0
Unfinished Stories are
User Story | Customer Priority | Developer Risk | Planned (hrs) | Actual (hrs) | Owner | Start Date | Acceptance Tests | Story Status
AddNewBooks | 3 | 1 | 20 | 0.0 | James Brown | 16/08/2002 | TestAddBooks: Not Done | Not Done | Edit | Delete
AddNewMembers | 0 | 1 | 15 | 0.0 | David Wilson | 15/08/2002 | TestAddMembers: Not Done | Not Done | Edit | Delete
Total number of unfinished stories=2
Done
Internet

For editing a particular story, click on the **Edit** button and to delete a story click on the **Delete** button. Once the edit button is clicked, the following fields of the user story can be edited. To edit a particular field, for example Story Description, click on the **Edit** button in the Story Description column.



Now edit the contents of the textbox and click on the **Submit** button. To view if the change has been made to the story description, click on "**Feature Management->View/Edit All Stories**" and select the story which had been previously edited.



Adding a new Task

In XP, a story may be split into one or more tasks and assigned to different team members. A user can add tasks by clicking on the "**Feature Management->Add Task**" menu item.

- Ensure that the project LibSoft is selected in the projects dropdown menu
- Enter new task-identifier, which is an alphanumeric string without spaces and special characters

  Task Identifier (alpha-numeric only): CreateBooksDatabase

- Enter the starting date for the user story in mm/dd/yyyy format

  Task Starting Date (MM/DD/YYYY): 08/20/2002

- Select the User Story to which the task belongs

  Choose a User Story: AddNewBooks

- Enter the task description

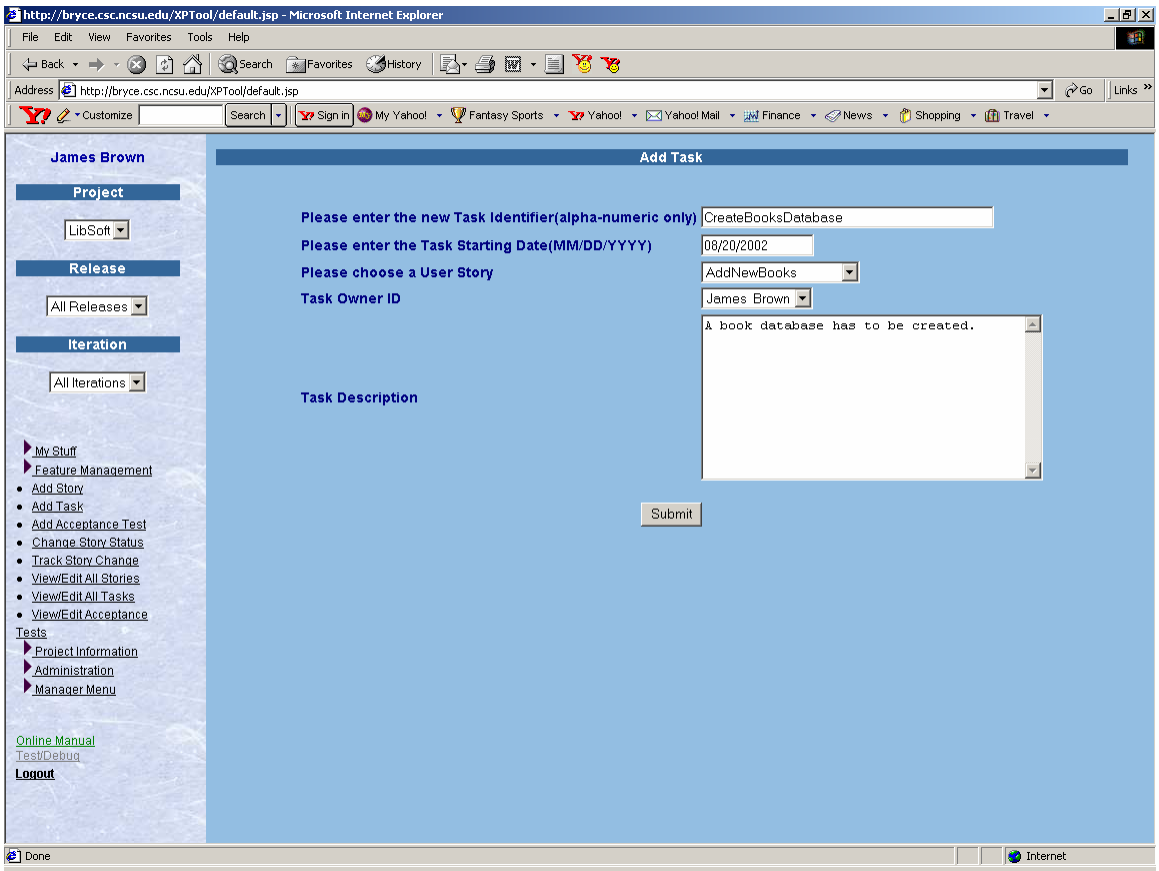
  Description: A Book database has to be created.

- Finally assign the task to a team member

  Task Owner ID: James Brown

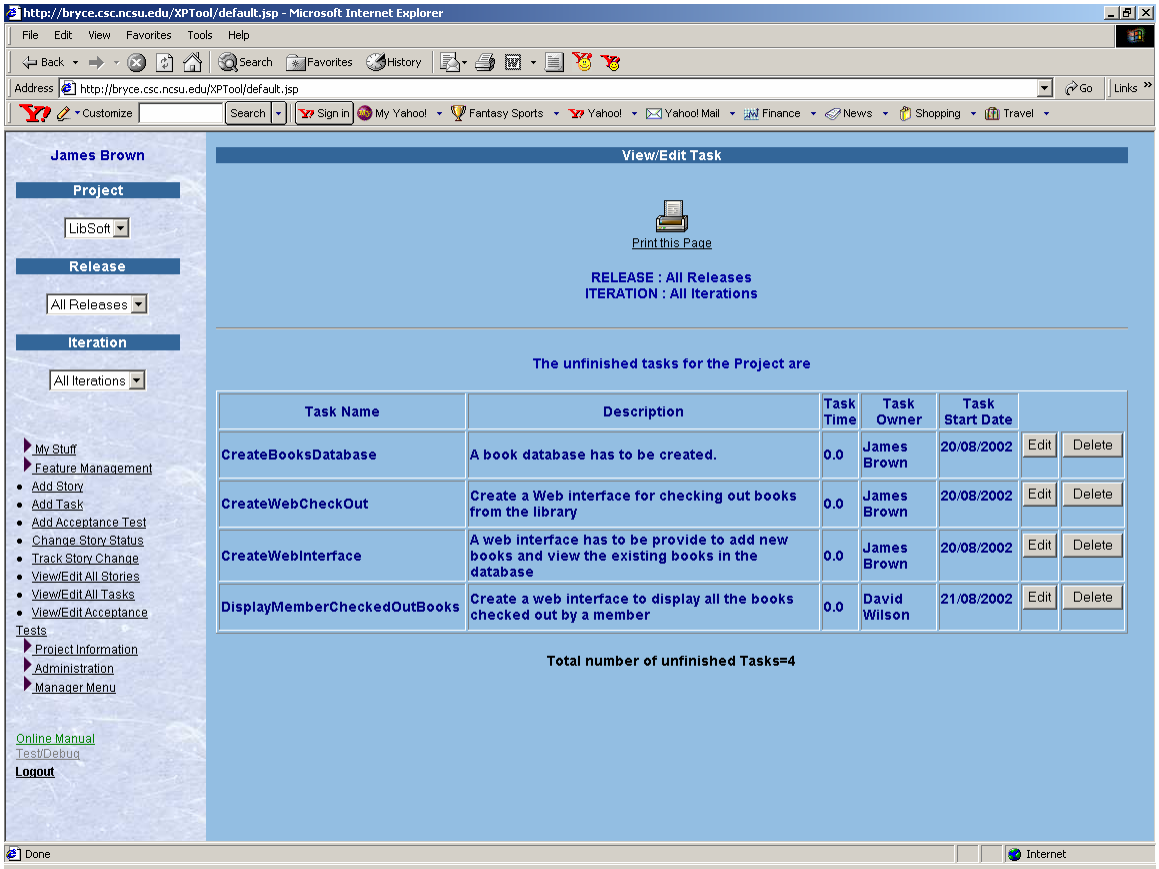
If the user is unsure of what to enter or has made a mistake when entering the data, he/she can change them later using "**Feature Management->View/Edit All Tasks**" menu item.

The following screen shows how to add CreateBooksDatabase user task.



Viewing and editing all Tasks

- For viewing all tasks in all releases and iterations, select “All Releases” in the release drop down menu and “All Iterations” from the iteration drop down menu. If the tasks of a particular Release or Iteration are to be viewed, select the particular Release and Iteration from the drop down menu.
- Now click on “**Feature Management->View/Edit All Tasks**” menu item


http://bryce.csc.ncsu.edu/XPTool/default.jsp - Microsoft Internet Explorer
James Brown
Project
LibSoft
Release
All Releases
Iteration
All Iterations
My Stuff
Feature Management
Add Story
Add Task
Add Acceptance Test
Change Story Status
Track Story Change
View/Edit All Stories
View/Edit All Tasks
View/Edit Acceptance Tests
Project Information
Administration
Manager Menu
Online Manual
Test/Debug
Logout
View/Edit Task
Print this Page
RELEASE : All Releases
ITERATION : All Iterations
The unfinished tasks for the Project are
Task Name | Description | Task Time | Task Owner | Task Start Date
CreateBooksDatabase | A book database has to be created. | 0.0 | James Brown | 20/08/2002 | Edit | Delete
CreateWebCheckOut | Create a Web interface for checking out books from the library | 0.0 | James Brown | 20/08/2002 | Edit | Delete
CreateWebInterface | A web interface has to be provide to add new books and view the existing books in the database | 0.0 | James Brown | 20/08/2002 | Edit | Delete
DisplayMemberCheckedOutBooks | Create a web interface to display all the books checked out by a member | 0.0 | David Wilson | 21/08/2002 | Edit | Delete
Total number of unfinished Tasks=4

For Editing / Deleting the Tasks, follow the same procedure as mentioned for the User stories.

Adding a new Acceptance Test

In XP, the customer writes an acceptance test to validate the user story. A user can add acceptance tests by clicking on the "**Feature Management->Add Acceptance Test**" menu item.

- Ensure that the project LibSoft is selected in the projects dropdown menu
- Enter new test-identifier, which is an alphanumeric string without spaces and special characters

  Test Identifier (alpha-numeric only): TestAddBooks
- Select the User Story to which the test belongs

  User Story: AddNewBooks
- Enter the test description, test file names, test expected results

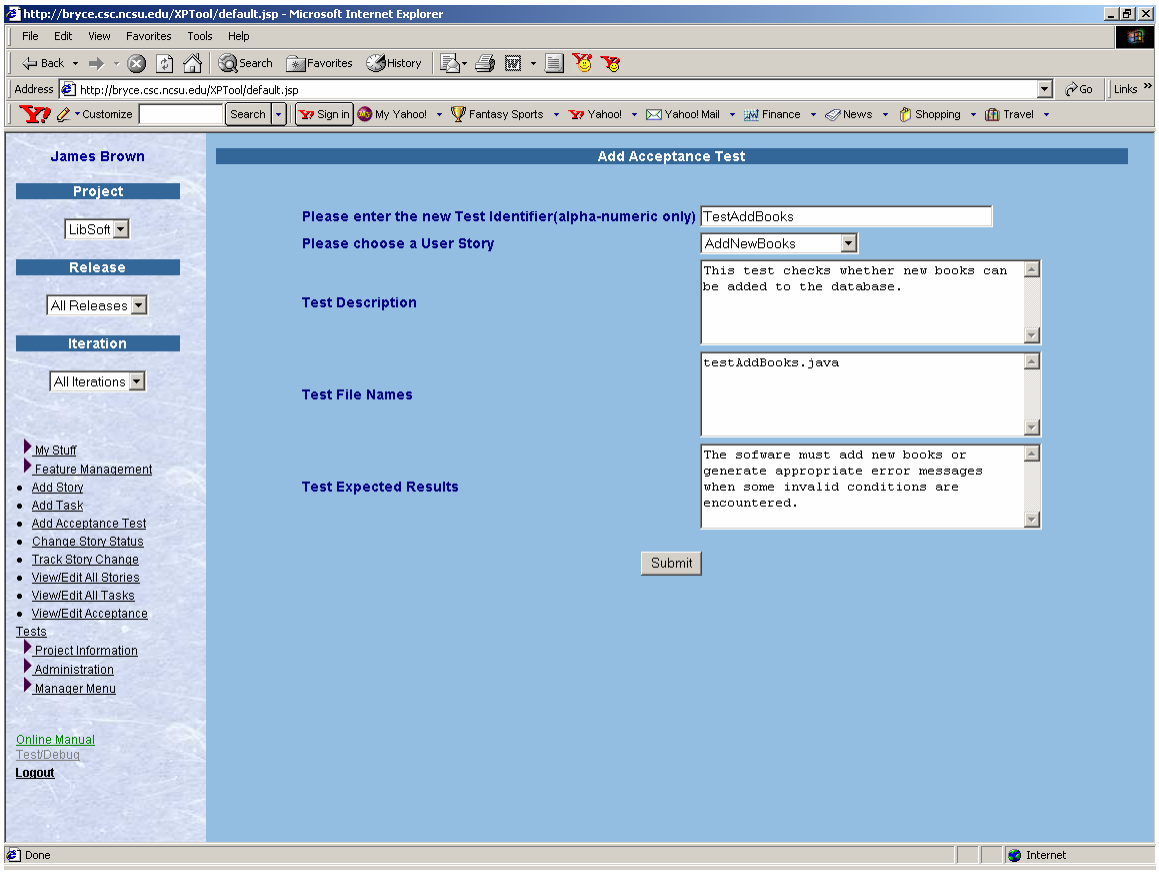
  Test Description: This test checks whether new books can be added to the database.

  Test File Names: testAddBooks.java

  Test Expected Results: The software must add new books or generate appropriate error messages when some invalid conditions are encountered.

If the user is unsure of what to enter or has made a mistake when entering the data, he/she can change them later using "**Feature Management->View/Edit Acceptance Tests**" menu item.

The following screen shows how to add the TestAddBooks acceptance test.

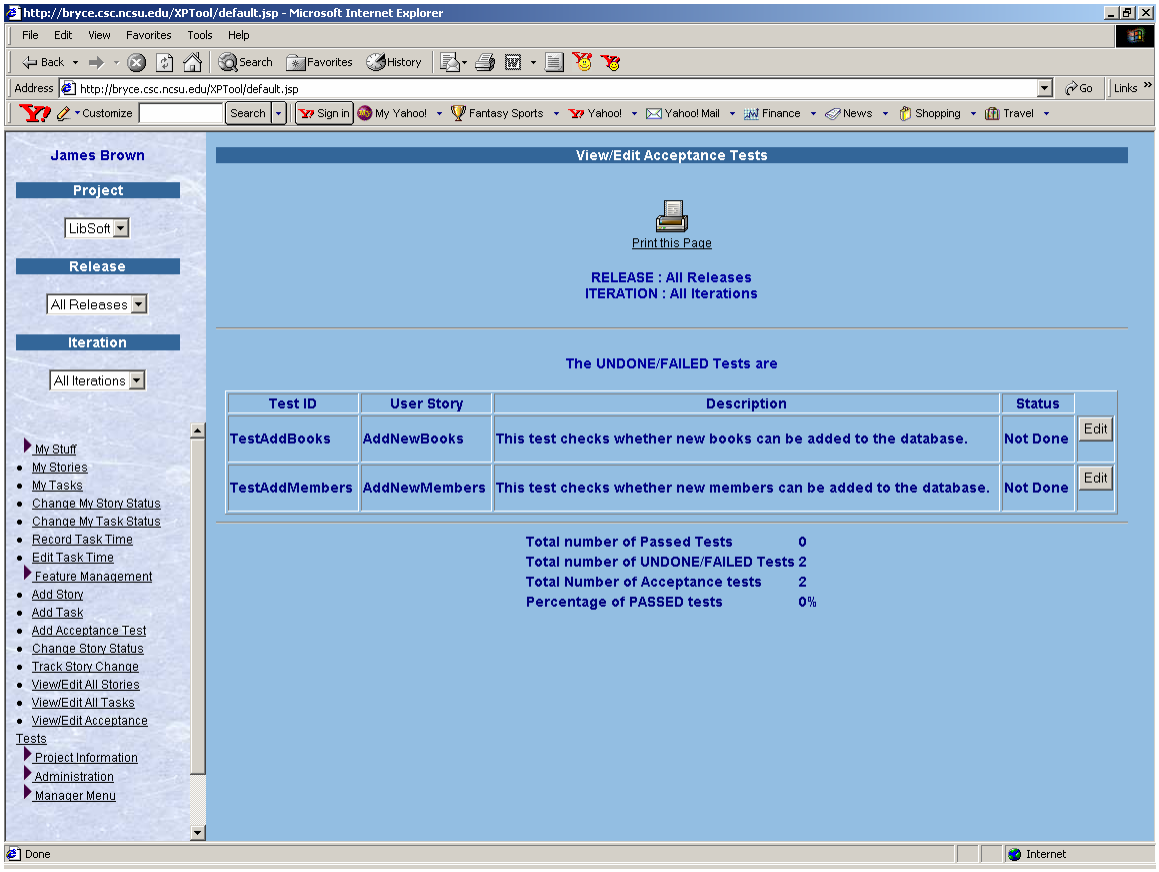


Similarly another test TestAddMembers can be added.

Viewing and editing all Acceptance Tests

- For viewing all tests in all releases and iterations, select “All Releases” in the release drop down menu and “All Iterations” from the iteration drop down menu. If the tests of a particular Release or Iteration are to be viewed, select the particular Release and Iteration from the drop down menu.

- Now click on "**Feature Management->View/Edit All Acceptance Tests**" menu item



Recording time spent on a Task

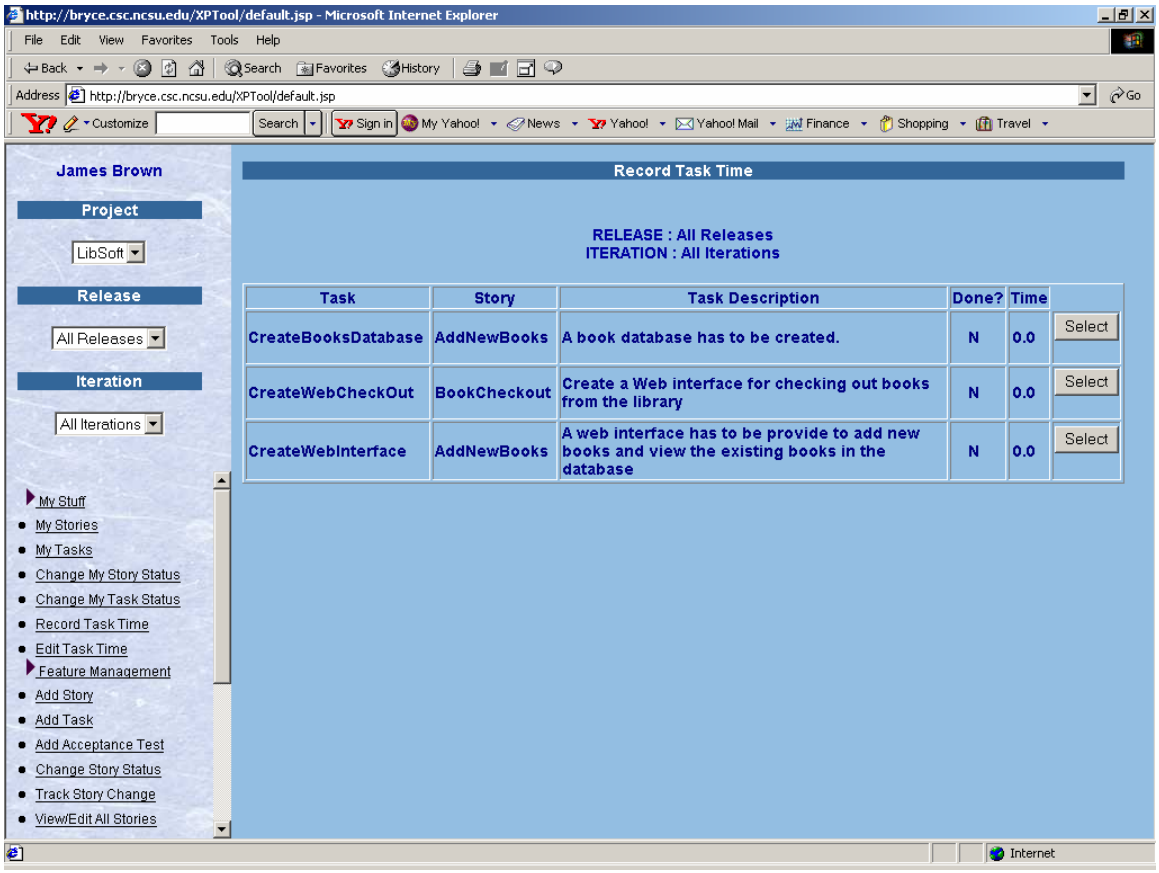
The tool captures time spent on a task via task sessions. A task session is a short duration of time spent implementing the task. There may be several sessions before a task can be complete. There are two types of sessions,

- Pair programming session – where two programmers collaborate and implement a task

- Solo programming session – where a programmer works alone on the task

Only the task owner can record time to tasks. The tool calculates the total time spent on a task by aggregating the time spent in all of its task sessions. In order to add a task session,

- List all tasks in all releases and iterations by selecting “All Releases” in the release drop down menu and “All Iterations” from the iteration drop down menu. If the tasks of a particular Release or Iteration are to be viewed, select the particular Release and Iteration from the drop down menu.
- Now click on “**My Stuff->Record Task Time**” menu item. In order to add a task session to the task CreateBooksDatabase, click on the corresponding select button.



Once the “Select” button is clicked the following page is displayed

- Enter the time spent in the session

  Time Spent: 20

- Choose on of the following from the collaboration time drop down menu
  - Total Time – this means the time entered in the time spent field is the total time for the session
  - Time per Person – this means that the time entered in the time spent field is multiplied by the number of collaborators (who are checked in the team member list that is displayed below)

  Choose: “Total Time“

- Enter the session date

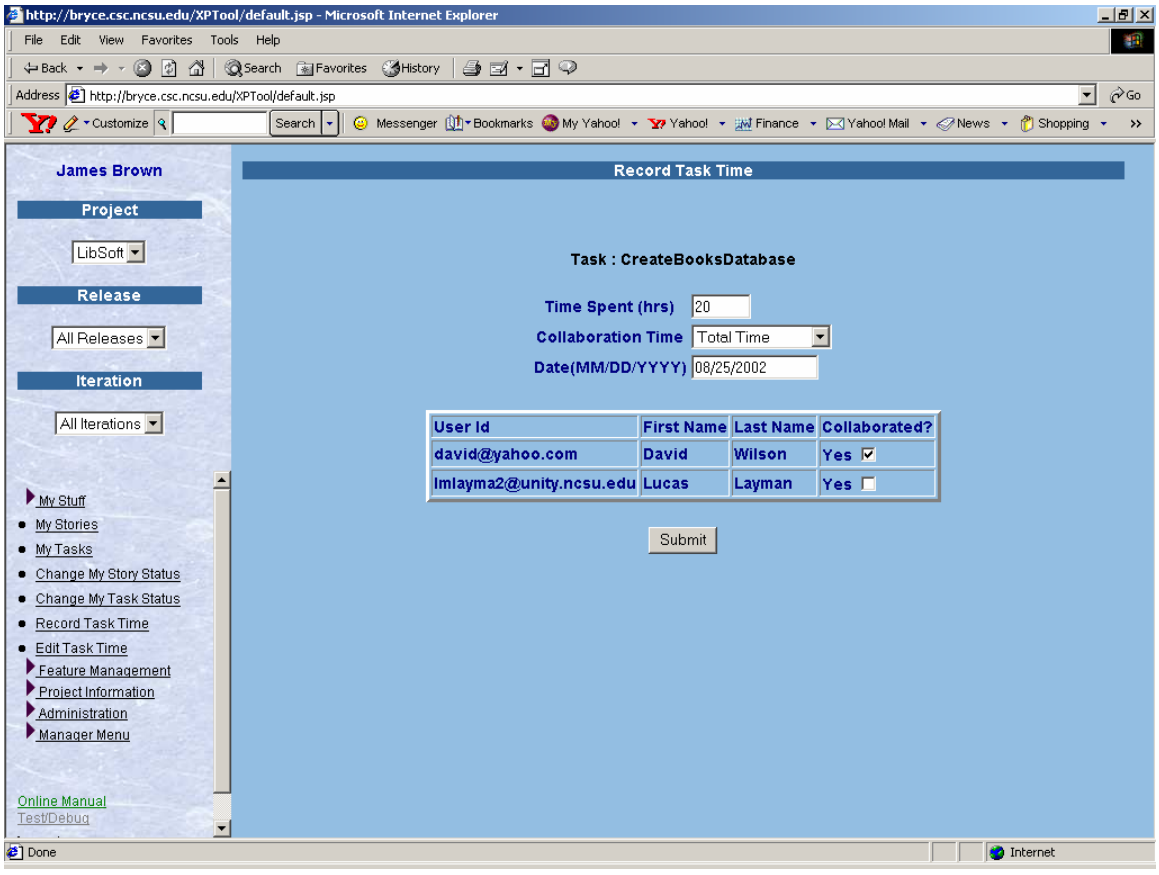
  Date (MM/DD/YYYY): 08/25/2002

- Check on the collaborator id with whom the user collaborated

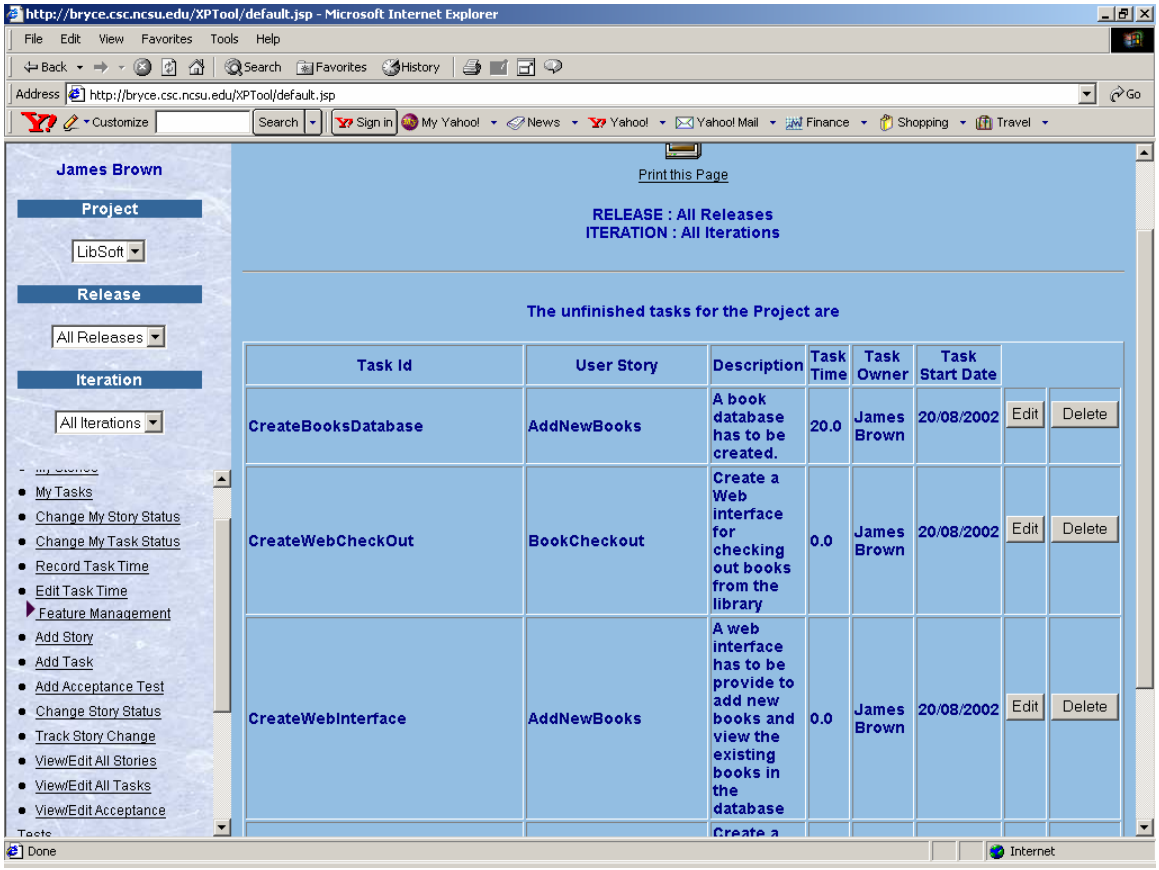
  User Id: David Wilson

- Click on the submit button

The task session for the task CreateBooksDatabase is as shown below. In this tutorial, the following information was entered in this page and the submit button was clicked.



After adding time to a task, using "**My Stuff->Record Task Time**", it can be viewed by clicking "**Feature Management->View/Edit All Tasks**". Notice that the Task time for CreateBookDatabase is now 20.

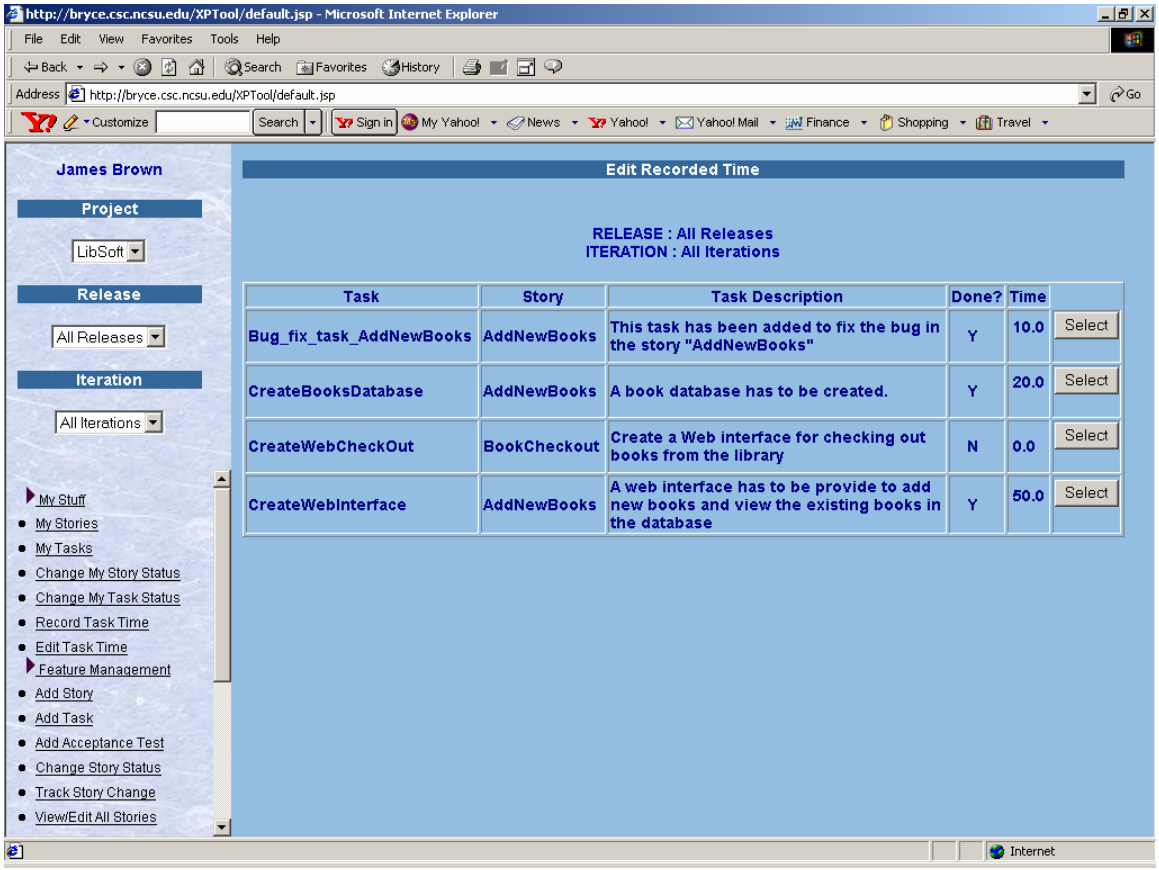


Editing the Recorded Task Time

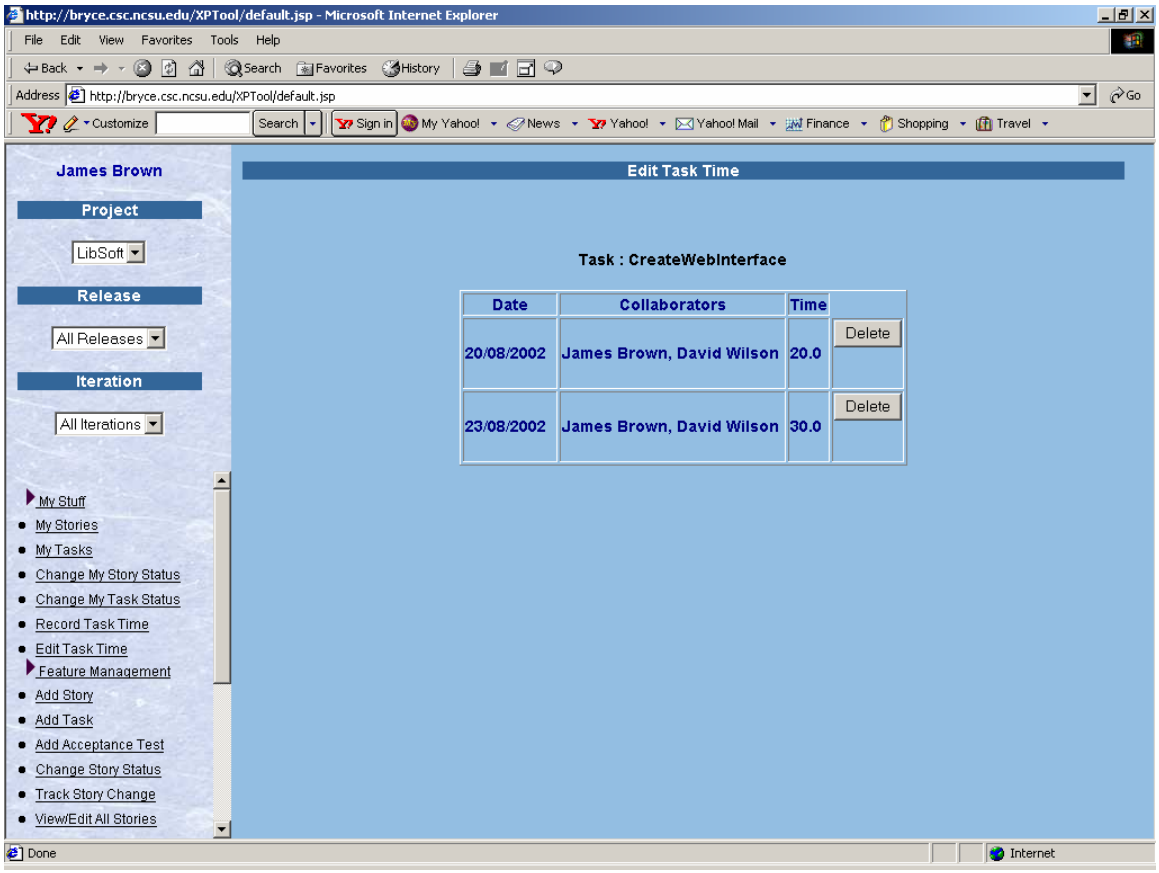
- For viewing all your tasks in all releases and iterations, select "All Releases" in the release drop down menu and "All Iterations" from the iteration drop down menu. If the tasks of a particular Release or Iteration are to be viewed, select the particular Release and Iteration from the drop down menu.

- Now click on “**My Stuff->Edit Task Time**” menu item

The following page is displayed. If the user wants to edit the task time of CreateWebInterface task, he/she has to click on the corresponding “Select” button.



Once the user has clicked on the Select button, the following page is displayed. It shows two task sessions, the collaborators for each session and the time spent in each session. If a user wishes to delete a session, he/she can click on the corresponding “Delete” button. To add a new session, click on “**My Stuff->Record Task time**” menu item.



Changing Task status to completed

Once the task is complete, its status can be changed from “not completed” to “completed”.

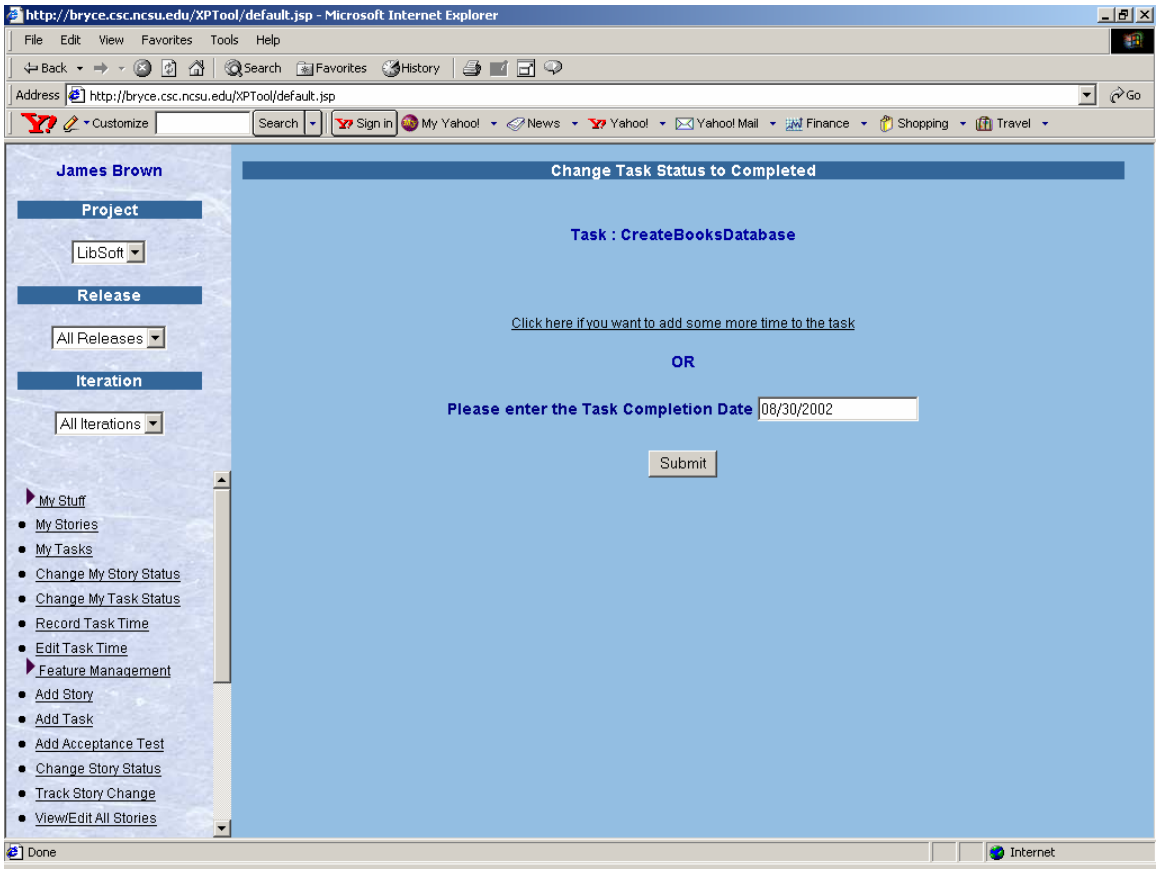
In order to change a task status,

- List all tasks in all releases and iterations by selecting “All Releases” in the release drop down menu and “All Iterations” from the iteration drop down menu. If the tasks of a particular Release or Iteration are to be viewed, select the particular Release and Iteration from the drop down menu.

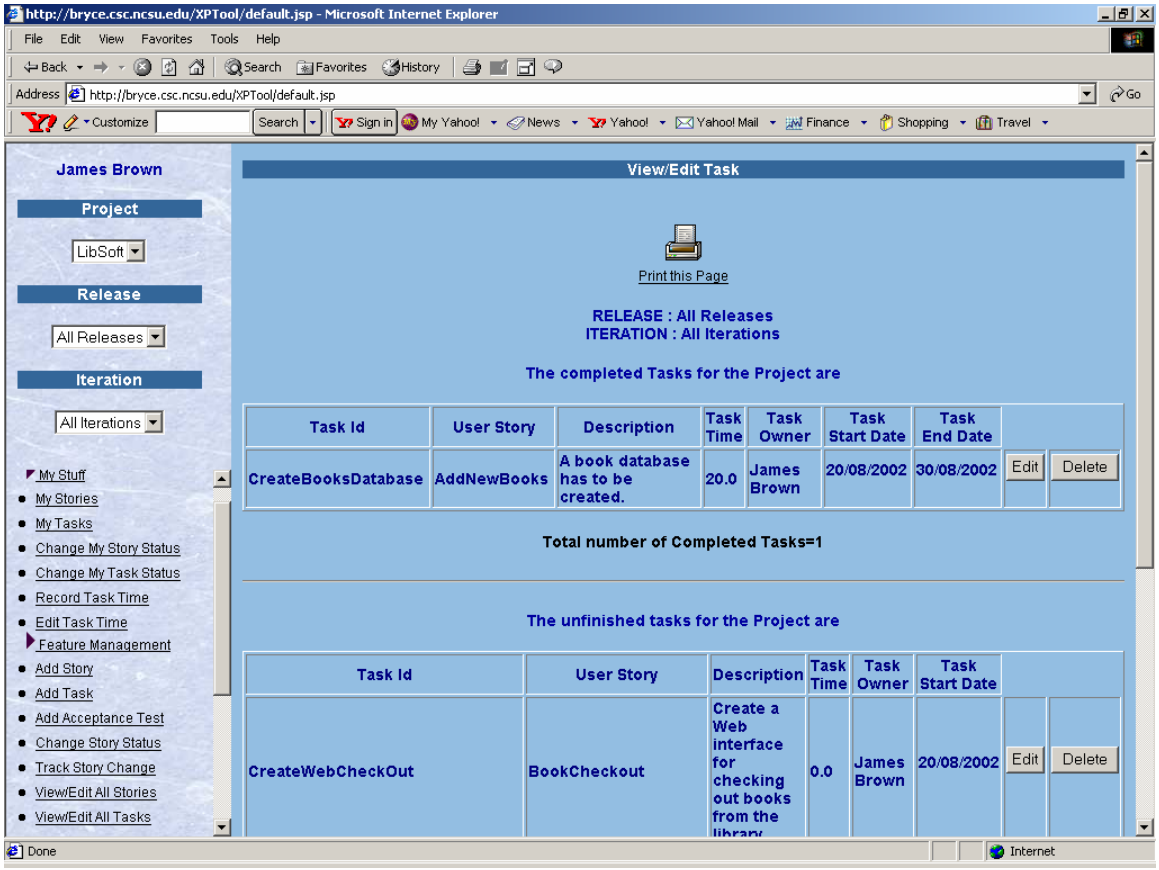
- Now click on "**My Stuff->Change My Task Status**" menu item. In order to change the status of the task CreateBooksDatabase, click on the corresponding select button.

The user can add more time to the task by clicking on the "Click here if you want to add some more time to the task" link. Else he/she can enter the task completion date and click on the Submit button.

To set the task CreateBooksDatabase to completed, enter 08/30/2002 in the task completion date field and click on the submit button.



After setting task status to completed, using “**My Stuff->Change My Task Status**”, it can be viewed by clicking “**Feature Management->View/Edit All Tasks**”. Notice that the task CreateBookDatabase is now listed as completed.


http://bryce.csc.ncsu.edu/XPTool/default.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://bryce.csc.ncsu.edu/XPTool/default.jsp
James Brown
Project
LibSoft
Release
All Releases
Iteration
All Iterations
My Stuff
My Stories
My Tasks
Change My Story Status
Change My Task Status
Record Task Time
Edit Task Time
Feature Management
Add Story
Add Task
Add Acceptance Test
Change Story Status
Track Story Change
View/Edit All Stories
View/Edit All Tasks
View/Edit Task
Print this Page
RELEASE : All Releases
ITERATION : All Iterations
The completed Tasks for the Project are
Task Id | User Story | Description | Task Time | Task Owner | Task Start Date | Task End Date
CreateBooksDatabase | AddNewBooks | A book database has to be created. | 20.0 | James Brown | 20/08/2002 | 30/08/2002 | Edit | Delete
Total number of Completed Tasks=1
The unfinished tasks for the Project are
Task Id | User Story | Description | Task Time | Task Owner | Task Start Date
CreateWebCheckOut | BookCheckout | Create a Web interface for checking out books from the | 0.0 | James Brown | 20/08/2002 | Edit | Delete
Done
Internet

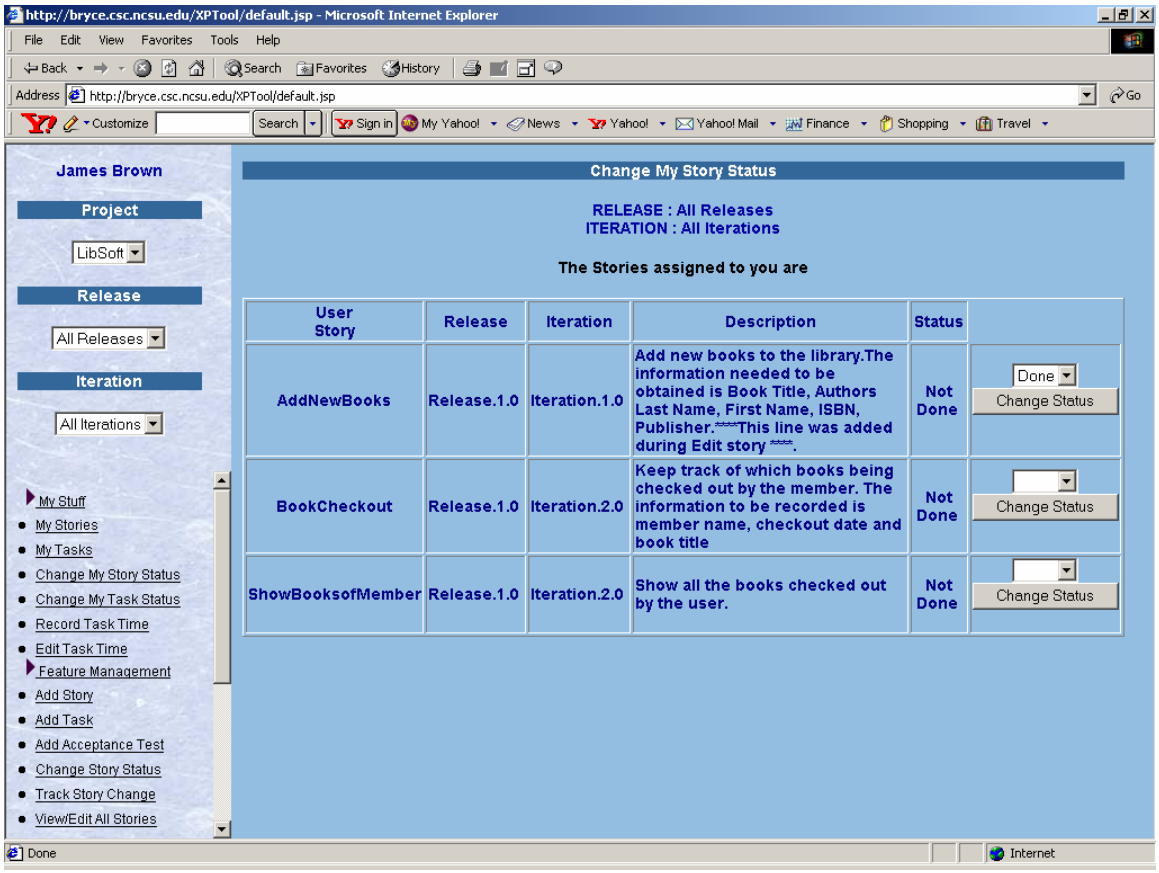
Changing Story status to completed

Once the Story is completed, its status can be changed from “Not Done” to “Done”.

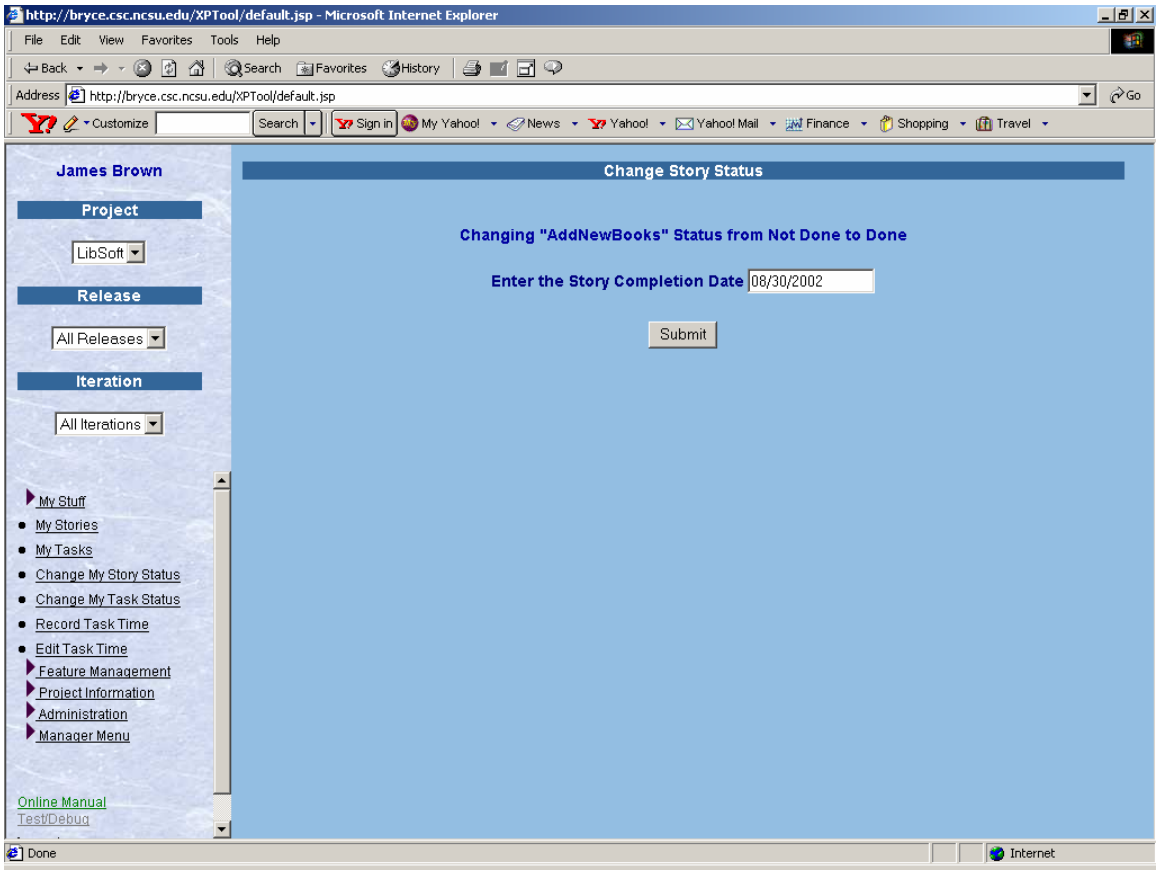
All the tasks of the user story must be completed before the story status can be changed to “Done” state.

In order to change a Story status,

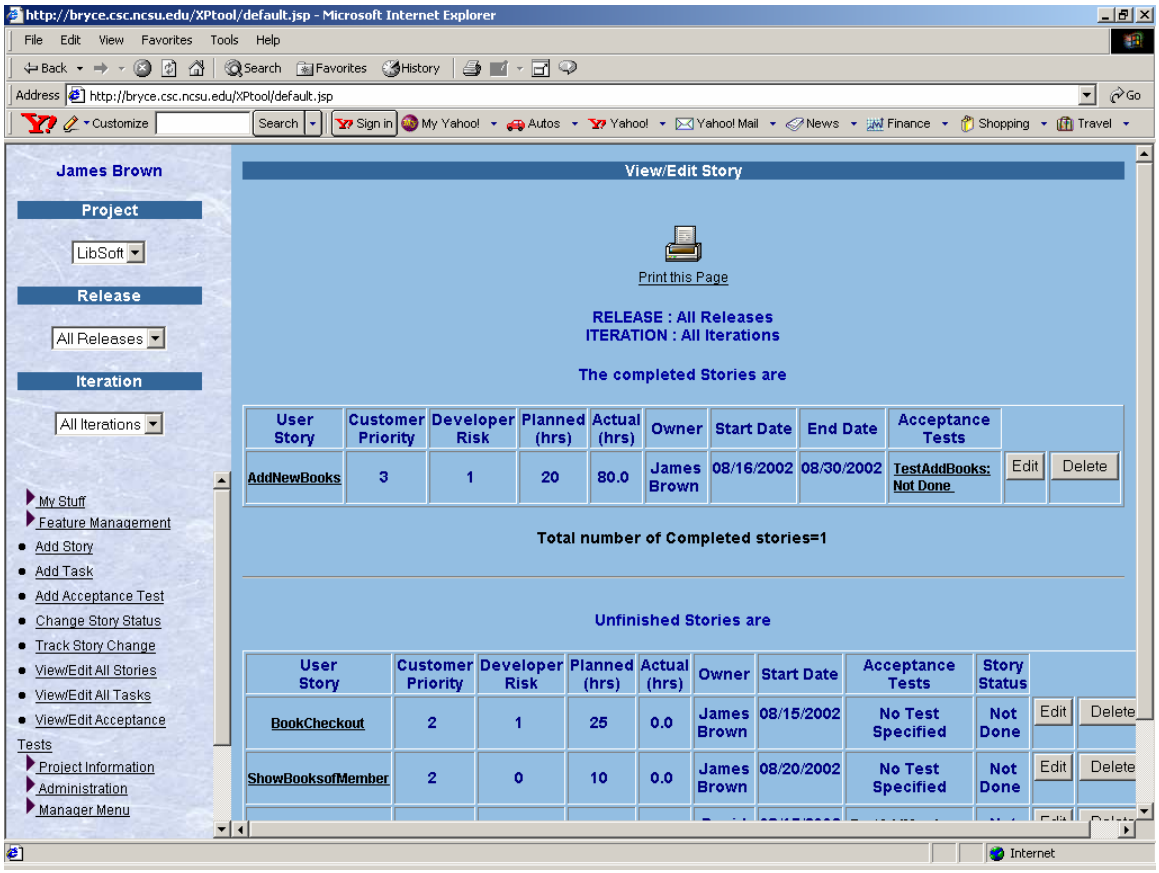
- List all stories in all releases and iterations by selecting “All Releases” in the release drop down menu and “All Iterations” from the iteration drop down menu. If the stories of a particular Release or Iteration are to be viewed, select the particular Release and Iteration from the drop down menu.
- Now click on “**My Stuff->Change My Story Status**” menu item. In order to change the status of the Story AddNewBooks, choose a new status “Done” from the drop down menu (Right now only one state “Done” is present in the drop down menu. Later we will see more user story states in the next section) and then click on the corresponding “Change Status” button.



If all the tasks of the user story AddNewBooks are completed, the story status can be changed to “Done” state and the following page is displayed. In the following page, enter the story completion date and click on the submit button.



After setting Story status to “Done”, using “**My Stuff->Change My Story Status**”, it can be viewed by clicking “**Feature Management->View/Edit All Stories**”. Notice that the Story AddNewBooks is now listed as completed.



Making changes to a completed Story

Once a user story has been completed, some changes may be made to it due to one of the following reasons

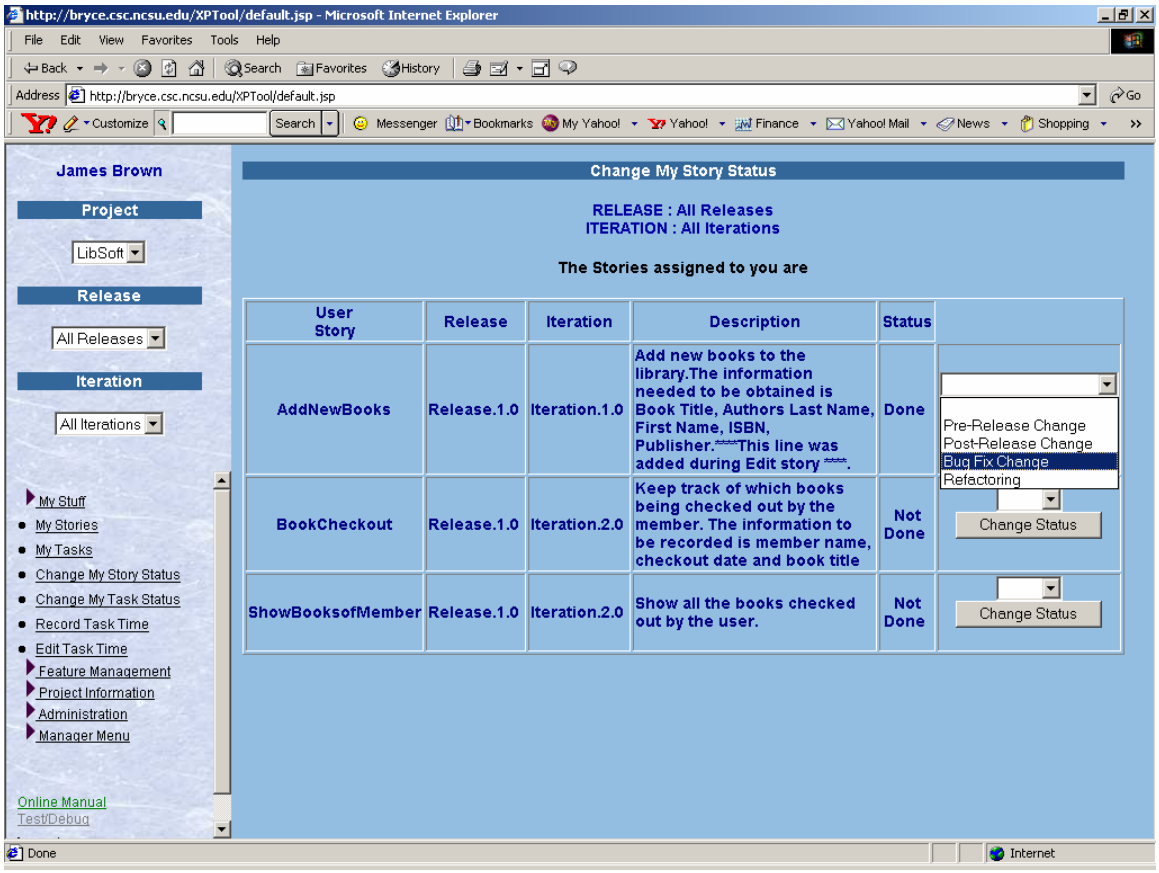
- Pre-Release Change – This change in the user story is requested by the customer prior to a release
- Post Release Change - This change in the user story is requested by the customer after a release
- Bug Fix Change – This change is required to fix a bug that has been found in the user story
- Refactoring – This change is made for refactoring the code.

In the tool, the story status can be changed from " Done" to "Pre-Release Change", "Bug Fix Change", "Refactoring" and "Post Release Change". When a Story enters one of the above four states, a new task is automatically added in order to make the change. This task has to be completed and then the user story can be changed back to "Done" state.

Reminder: All the tasks of the user story must be completed before the story status can be changed to "Done" state.

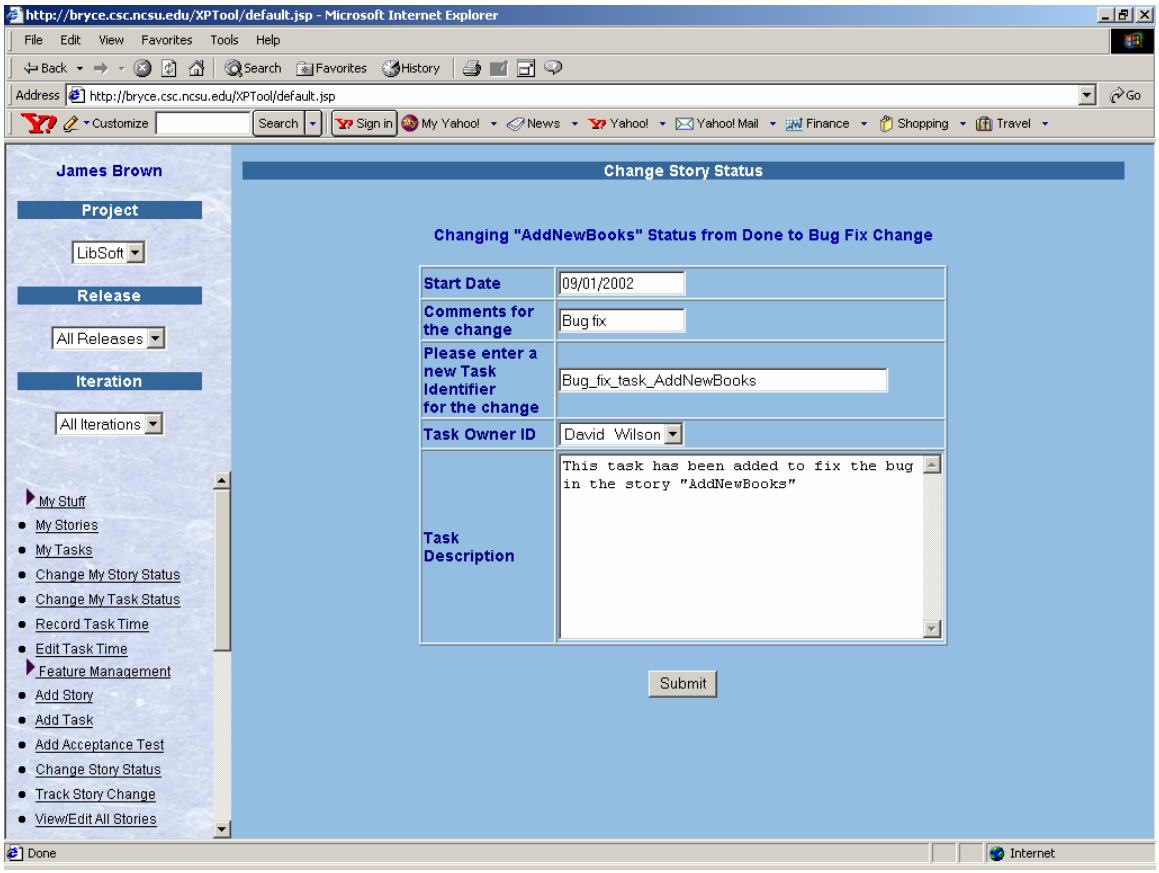
In order to change a Story status,

- List all stories in all releases and iterations by selecting "All Releases" in the release drop down menu and "All Iterations" from the iteration drop down menu. If the stories of a particular Release or Iteration are to be viewed, select the particular Release and Iteration from the drop down menu.
- Now click on "**My Stuff->Change My Story Status**" menu item. In order to change the status of the Story AddNewBooks form "Done" to "Bug Fix Change", choose a new status "Bug Fix Change" from the drop down menu and then click on the corresponding "Change Status" button below it.



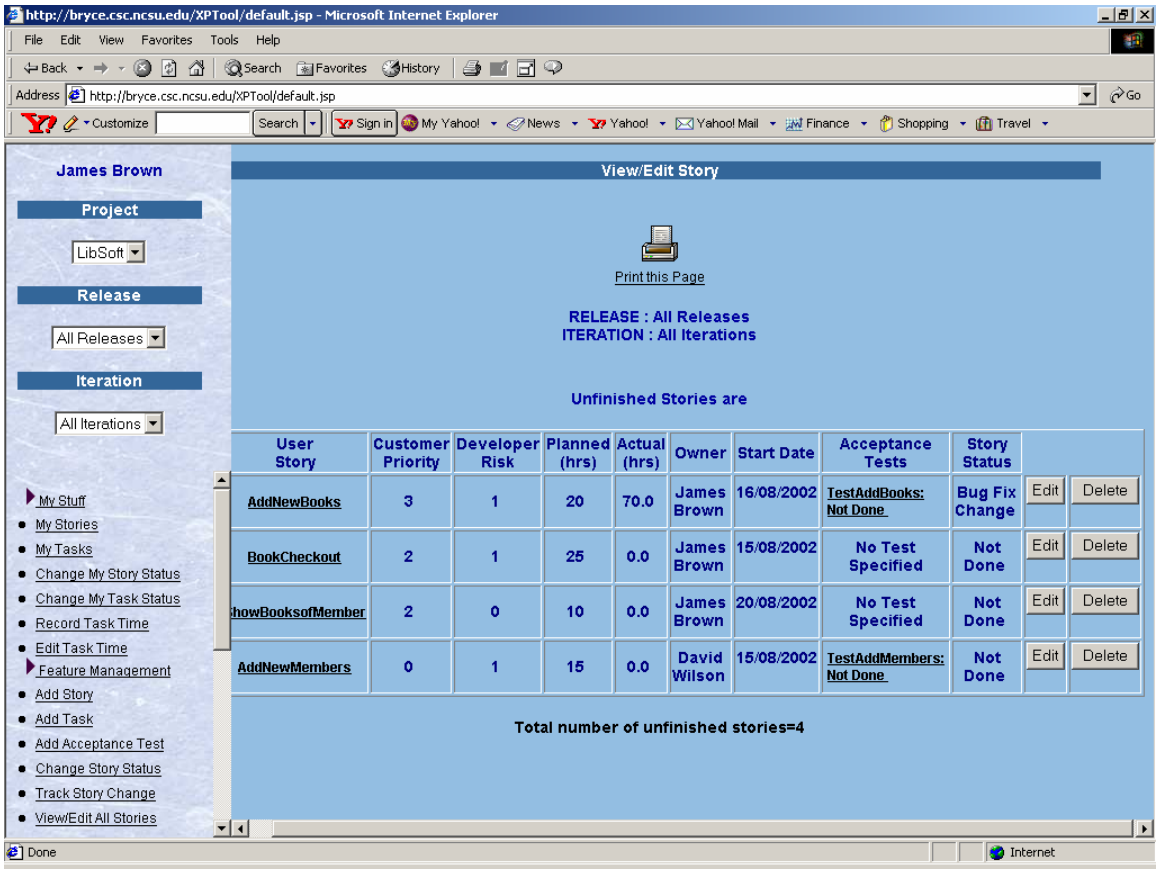
Once the “Change Status” button is clicked, the following page is displayed. The user is required to enter a new task for making this change. After entering the information, the user has to click on the “Submit” button.

The following page shows the user story AddNewBooks being changed to “Bug Fix Change” state. A new task Bug_fix_task_AddNewBooks is being added to make this change.



After clicking the submit button, the new story status can be viewed by clicking "**Feature Management->View/Edit All Stories**". Notice that the Story AddNewBooks Status is now "Bug Fix

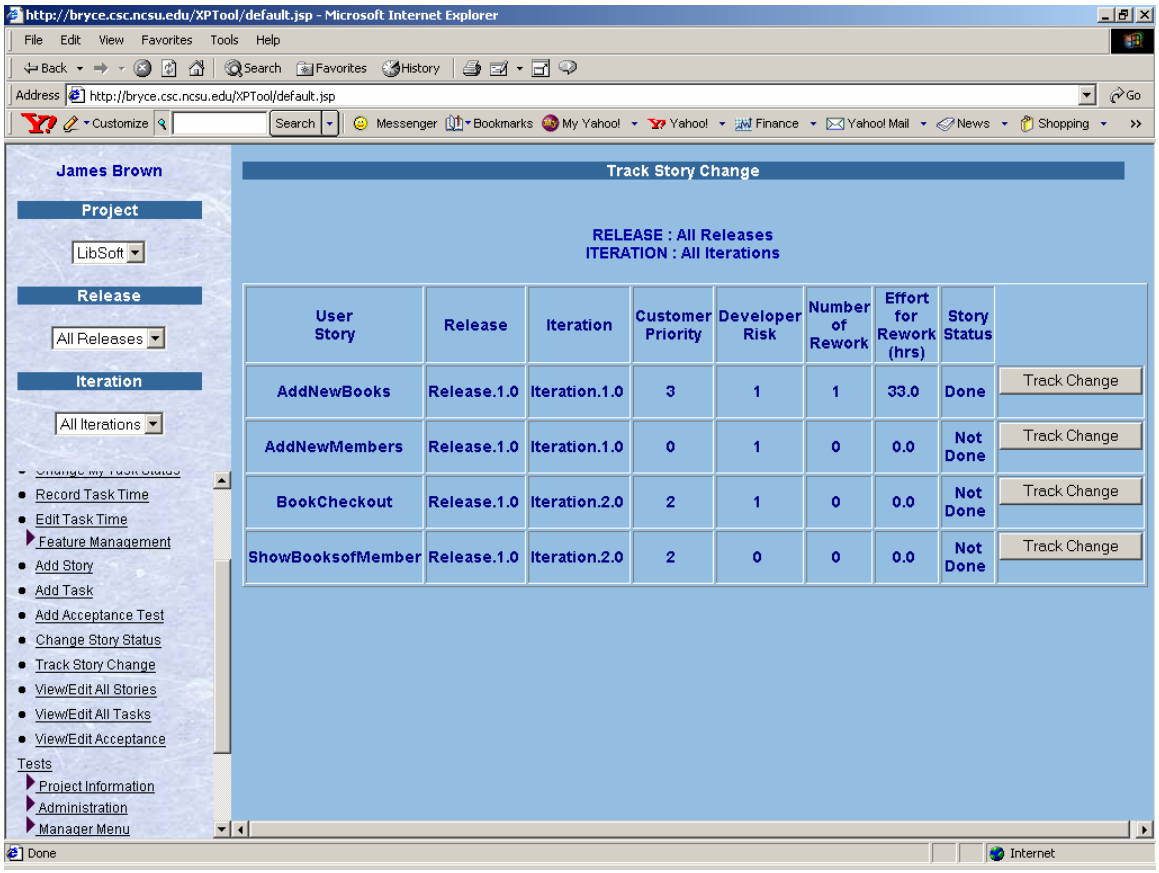
Change”.


http://bryce.csc.ncsu.edu/XPTool/default.jsp - Microsoft Internet Explorer
James Brown
Project
LibSoft
Release
All Releases
Iteration
All Iterations
My Stuff
My Stories
My Tasks
Change My Story Status
Change My Task Status
Record Task Time
Edit Task Time
Feature Management
Add Story
Add Task
Add Acceptance Test
Change Story Status
Track Story Change
View/Edit All Stories
View/Edit Story
Print this Page
RELEASE : All Releases
ITERATION : All Iterations
Unfinished Stories are
User Story | Customer Priority | Developer Risk | Planned (hrs) | Actual (hrs) | Owner | Start Date | Acceptance Tests | Story Status
AddNewBooks | 3 | 1 | 20 | 70.0 | James Brown | 16/08/2002 | TestAddBooks: Not Done | Bug Fix Change | Edit | Delete
BookCheckout | 2 | 1 | 25 | 0.0 | James Brown | 15/08/2002 | No Test Specified | Not Done | Edit | Delete
howBooksofMember | 2 | 0 | 10 | 0.0 | James Brown | 20/08/2002 | No Test Specified | Not Done | Edit | Delete
AddNewMembers | 0 | 1 | 15 | 0.0 | David Wilson | 15/08/2002 | TestAddMembers: Not Done | Not Done | Edit | Delete
Total number of unfinished stories=4
Done
Internet

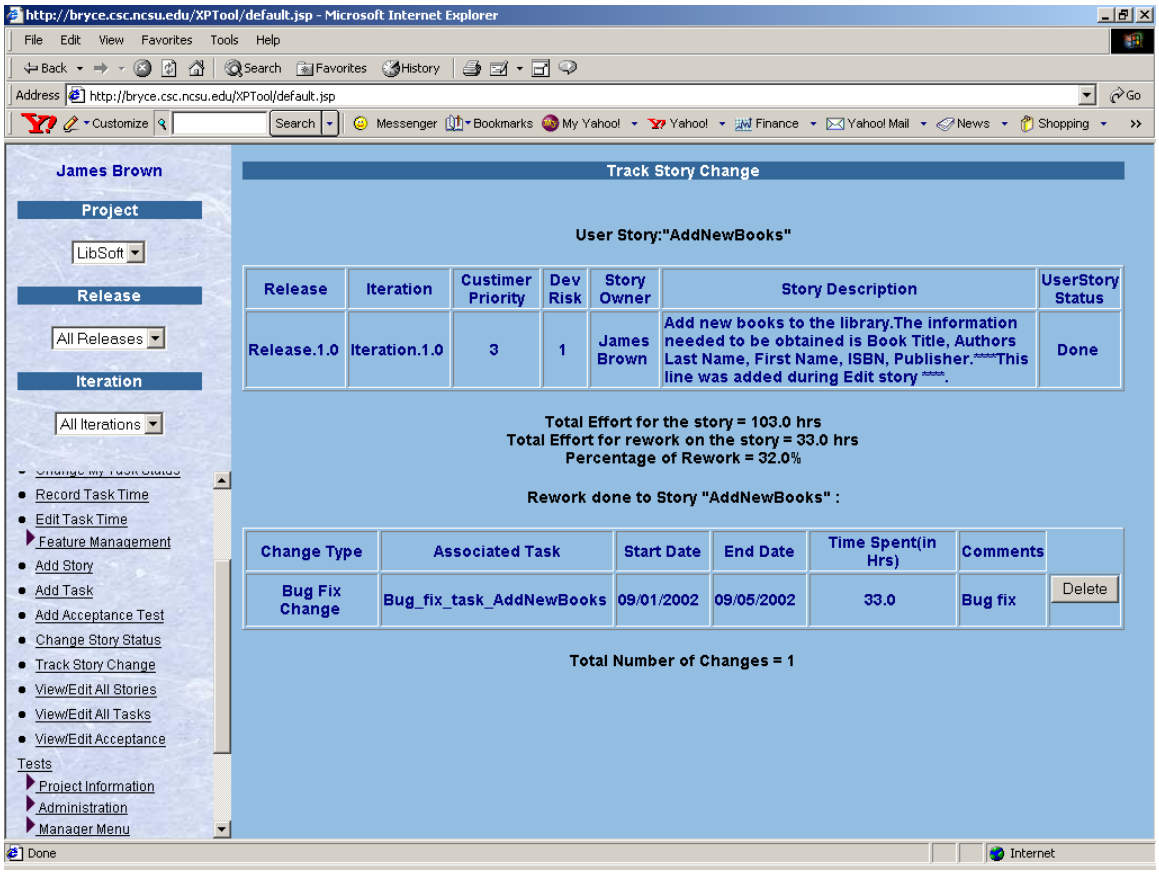
Keeping track of changes made to a story

- For listing all Stories in all releases and iterations, select “All Releases” in the release drop down menu and “All Iterations” from the iteration drop down menu. If the stories of a particular Release or Iteration are to be viewed, select the particular Release and Iteration from the drop down menu.
- Now click on “**Feature Management->Track Story**” menu item

  The following page is displayed. To view the changes made to a story AddNewBooks, click on the corresponding “Track Change” button.



The following page lists the changes that have been made to the story AddNewBooks



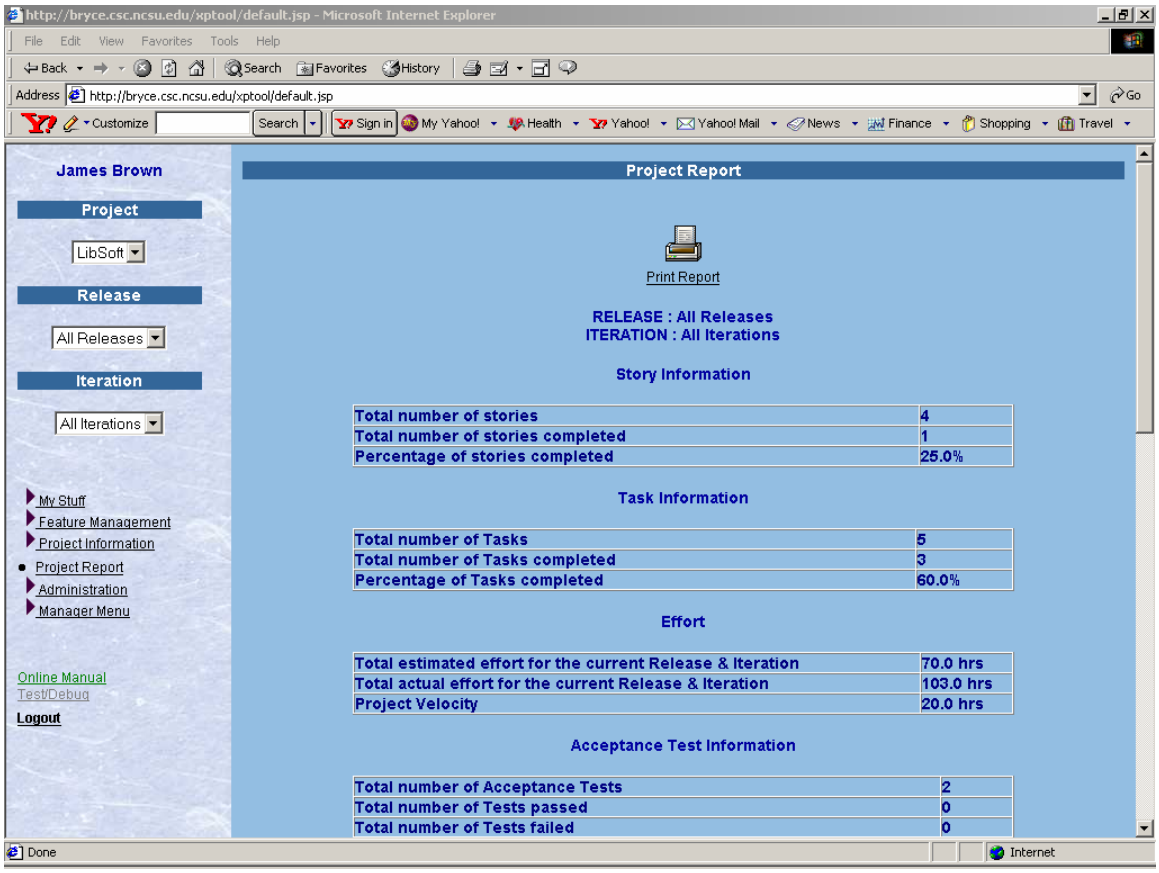
To delete the change, click on the “Delete” button corresponding to the change.

Viewing Project Report

In order to view the project report,

- For the aggregate project report of all releases and iterations, select “All Releases” in the release drop down menu and “All Iterations” from the iteration drop down menu. For a particular Release or Iteration, select the particular Release and Iteration from the drop down menu.
- Now click on “**Project Information->Project Report**” menu item.

The project report of LibSoft for all releases and iterations is as shown below



Tool Administration

This menu item is used for the administration of the project.

Add Team Member

This submenu item allows a normal user (i.e. not a manager) to add a new team member to the project. The new team member must have registered to the system before he can be added as a team member to the project. This menu item displays the current team members of the project. To add a new team

member, the username and the password must be entered. The user is required to enter the following fields:

- User ID (Email id used during registration)
- Password

Delete User Registration

This submenu item is used to delete a user from the system. All the information pertaining to the user is destroyed. This menu item deletes the currently logged in user from the system. To delete a user, the password must be entered.

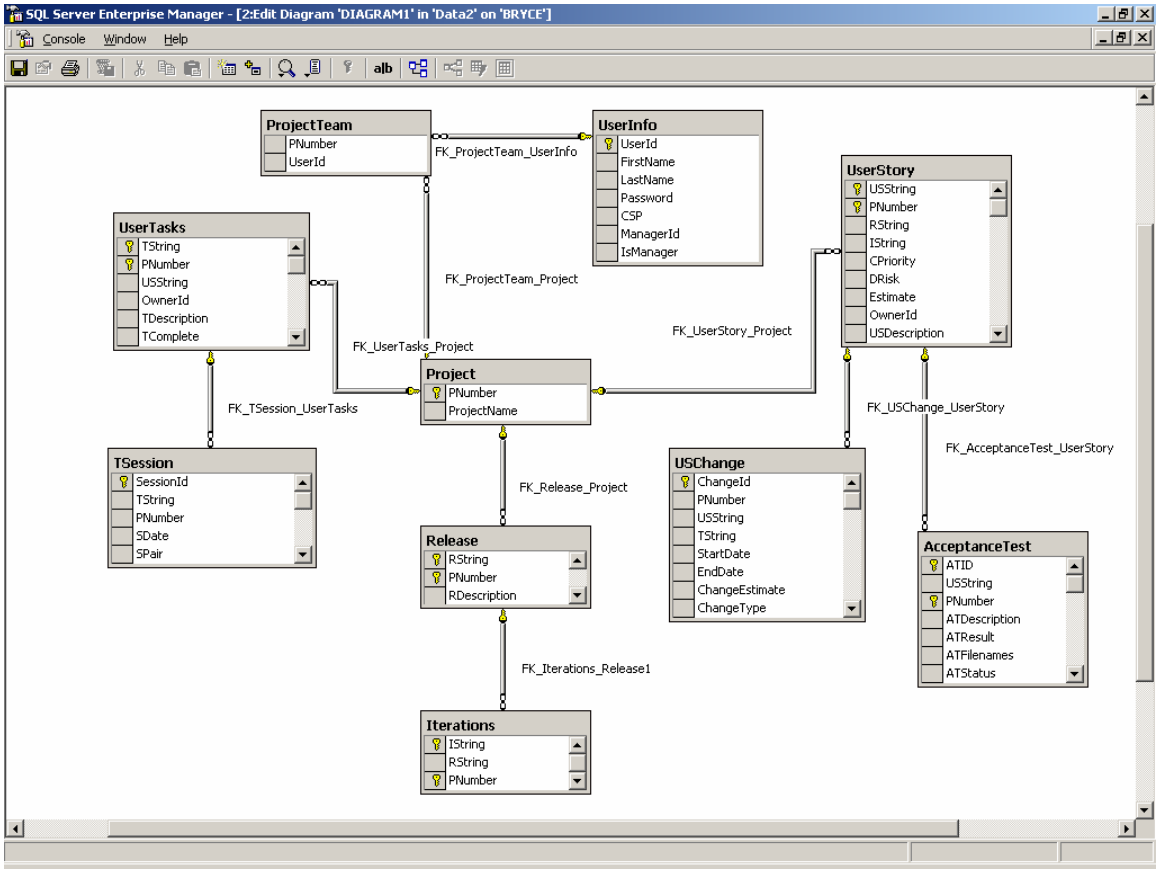
Add/Change Manager

This submenu item is used to change the current project manager. The user can choose a new manager from a list of project managers registered to the system.

Change Password

This submenu item is used to change the password of the current user.

APPENDIX 4: Database Diagram



APPENDIX 5: Bryce Survey

Thanks for using Bryce.

This survey is for the new version of Bryce (XP Project Management Tool).

We would like your suggestions so that we can refine it further.

Please note that this information will be confidential.

Personal Information

- Enter your email address (UserId) (This information is confidential and is used for information purposes only)
- User Group
    - Answer Choices - CSC326, ECE463, ECE521, CSC517, IBM, Genesys

Project Management

- You liked implementing the project using the XP methodology
    - Answer Choices - Strongly Agree, Agree, Disagree, Strongly Disagree
- Have you used a project management tool before?
    - Answer Choices – No, Yes
- How does your team work?
    - Answer Choices – Distributed, Colocated
- Did your team practice pair programming?
    - Answer Choices – No, Yes

- Accessing project information online was useful to you
    - Answer Choices - Strongly Agree, Agree, Disagree, Strongly Disagree
- You tracked/monitored the project progress using the tool
    - Answer Choices - All the time, Sometimes, Never
- The tool can save time and/or effort in tracking the project
    - Answer Choices - Strongly Agree, Agree, Disagree, Strongly Disagree
- The project report (that is generated automatically) is useful
    - Answer Choices - Strongly Agree, Agree, Disagree, Strongly Disagree
- How many team members entered data or monitored the project in the tool?
    - Answer Choices - Everyone on the Team, Most Team Members, Few Team Members, One Member
- The tool is a better alternative to story cards (i.e. having requirements on index cards)
    - Answer Choices - Strongly Agree, Agree, Disagree, Strongly Disagree
- The tool can be useful in industrial projects
    - Answer Choices - Strongly Agree, Agree, Disagree, Strongly Disagree

About Stability

- Were you able to access the tool all the time i.e. was it stable?
    - Answer Choices - All the time, Most of the time, Sometimes, Never
- How many bugs are did you find in the tool?
    - Answer Choices - 0 Bugs, 1-2 Bugs, 3-5 Bugs, 6-10 Bugs, More than 10 Bugs

About Learning, Usage and Documentation

- The tool is simple to use
    - Answer Choices - Strongly Agree, Agree, Disagree, Strongly Disagree
- How many minutes/week do you think you spend updating Bryce?
    - Answer Choices - 0-15 minutes, 16-30 minutes, 30-45 minutes, More than 45 minutes
- Approximately how long did it take to learn the tool (using the tutorial/Manual etc)?
    - Answer Choices - 1 hr, Few hrs, 1 day, Few days, A Week or more
- The tutorial and user manual were helpful in learning the tool
    - Answer Choices - Strongly Agree, Agree, Disagree, Strongly Disagree

Overview

- Please comment on your overall impression and experience in using the tool

- Please list any features you would like to add to the tool, bug information or any suggestions you have for making the tool easier to use/navigate
- Any other comments